

1N-61-12
0.17

**THE CATHOLIC UNIVERSITY OF AMERICA
DEPARTMENT OF ELECTRICAL ENGINEERING**

ANNUAL REPORT

ON

**Study and Analysis of
The Robot-Operated Material Processing Systems (ROMPS)**

Research Grant NAG 5-780

Dr. Charles C. Nguyen
Principal Investigator and Professor

and

Hui-I Liu
Graduate Research Assistant

Submitted to

Mr. Lloyd Purves
Code 712
Technical Officer
Goddard Space Flight Center (NASA)
Greenbelt, Maryland

March 1996

REPORT SUMMARY

This is a report presenting the progress of a research grant funded by NASA for work performed from October 1, 1994 to September 31, 1995. The report deals with the development and investigation of potential use of software for data processing for the Robot Operated Material Processing System (ROMPS). It reports on the progress of data processing of calibration samples processed by ROMPS in space and on earth. First data were retrieved using the I/O software and manually processed using Microsoft Excel. Then the data retrieval and processing process was automated using a program written in C which is able to read the telemetry data and produce plots of time responses of sample temperatures and other desired variables. LabView was also employed to automatically retrieve and process the telemetry data.

1. Introduction

The Robot Operated Materials Processing System (ROMPS) has been developed at Goddard Space Flight Center (GSFC) under a flight project to investigate commercially promising in-space material processes and to design reflyable robot automated systems to be used in the above processes for low-cost operations[1]. The ROMPS is currently scheduled for flight in 1994 as a Hitchhiker payload in a Get Away Special (GAS) can. An important component of the ROMPS is a three degree-of-freedom (DOF) robot which will be responsible for carrying out the required tasks of in-space processing of selected materials. The mechanical interfaces and the compliant device of the ROMPS robot were tested using a six degree of freedom (DOF) platform and the test results and recommendation can be found in [2]. This report deals with the processing of the data of the critical ROMPS variables such as sample temperatures and actuator motor currents, etc. Progress on data processing using various software packages is reported. The report is organized as follows. Section 2 reports work on the application of MicroSoft Excel for data processing. Results of the development of a program written in LabView to retrieve and process the telemetry data are presented in Section 3. Section 4 presents the development of a program written in C language that is able to retrieve, process the telemetry data and produce hard copy of the time responses of selected variables.

2. Data Processing Using Excel

The flight telemetry were converted to a worksheet compatible format (such as Quattro Pro, Lotus or MicroSoft Excel) using a developed I/O software program. The converted data were put on the ROMPS server that is located at Goddard Space Flight Center. From Catholic University, we used the ftp facility to retrieve the worshheet compatible format data from the ROMPS server. The retrieved data are then stored in an external drive (Drive G:) which has a storage capacity of about 3 giga bytes at a CUA personal computer. MicroSoft Excel was used to read the data and numerous plots were produced and described below. The plots produced in this section are obtained by manually entering the commands by the user into Excel. The task was very tedious and time-consuming. It is noted that the two flight furnaces are called Furnace A

and B. The furnace that is currently located at ITE and will be sent Dr. Tim Anderson at University of Florida is called Furnace C. The plots as shown in Appendix A are organized as follows:

Set 1- Flight Data: There are 14 plots obtained from the telemetry data. The plots illustrate the time responses (of temperatures from thermocouples C1 and C2 and the power profiles) of the first and second runs of Samples 44, 45, 62, 73, 74, 87, and 100. The temperatures are given in Celsius degrees and the power in centiwatts. The first run was performed on Furnace B and the second run on Furnace A.

Set 2-GSFC Data: There are 13 plots obtained from the tests made at the GSFC Optical Site on May 12, 1995. All the tests were performed on Furnace B. Seven plots illustrate the time responses of the first run of Samples 44, 45, 62, 73, 74, 87, and 100 and six plots the second run of Samples 44, 45, 62, 73, 74, and 87. We do not know why the second run was not performed for Sample 100.

Set 3-ITE Data: This set contains 26 plots obtained from the test data of study cases performed at ITE by Michelle and ITE personels. There are two subsets of plots. Subset 1 contains plots of time responses of the first and second runs for the *Up Study Case* in which the sample was facing up and the furnace was facing down. The samples that were tested in the Up Study Case are 44, 45, 62, 73, 74 and 87 resulting in a total of 12 plots. Subset 2 contains plots of time responses of the first and second runs for the *Down Study Case* in which the sample was facing down and the furnace was facing up. The samples that were tested in the Down Study Case are 44, 45, 62, 73, 74, 87 and 100 resulting in a total of 14 plots.

The plots from the flight data and GSFC data are represented on the same scale on both vertical and horizontal axes for comparison purpose. The sample, the run number and furnace ID are given on each of the above plots.

In addition to the "processed data" as shown in Appendix 1, it was requested from University of Florida that CUA process raw data and plot it together with the "processed" data. The results of this data processing are presented in Appendix 2 and are summarized as follows

For each of the calibration sample 44, 45 and 62 we have:

Plot 1: Processed Data of Power, Sensor 1 (Flight), Sensor 1 (Goddard) are plotted with respect to time. Both power curves from Flight and Goddard are for the purpose of lining up the time.

Plot 2: Processed Data of Power, Sensor 2 (Flight), Sensor 2 (Goddard) are plotted with respect to time.

Plot 3: Raw Data of Power, Sensor 1 (Flight), Sensor 1 (Goddard) are plotted with respect to time.

Plot 4: Raw data of Power, Sensor 2 (Flight), Sensor 2 (Goddard) are plotted with respect to time.

3. Data Processing Using LabView

LabView was used to retrieve the telemetry data without using the developed I/O program and to produce hard copy of the variable time responses on a laser printer. The diagram and front panel of the LabView program are given in Appendix 3.

4. Data Processing Using a C Program

As mentioned in Section 1, the task of manually entering commands to produce hard copies of variable time responses was very tedious and time-consuming. This has motivated the developement of a software package that can automatically retrieve and process the data. As a consequence, a program was written in C to facilitate the data processing. It is able to retrieve the telemetry data without using the I/O program. The C program provides its user with a selection menu of multiple choices. The user can either select to process the data of one single sample or the data of a range of samples. The C program displays the time response on the computer screen and then produce the hard copy of the plot on a laser printer. A listing of the source codes of the C program is given in Appendix 4 and some typical plots of the variable time responses are given in Appendix 5.

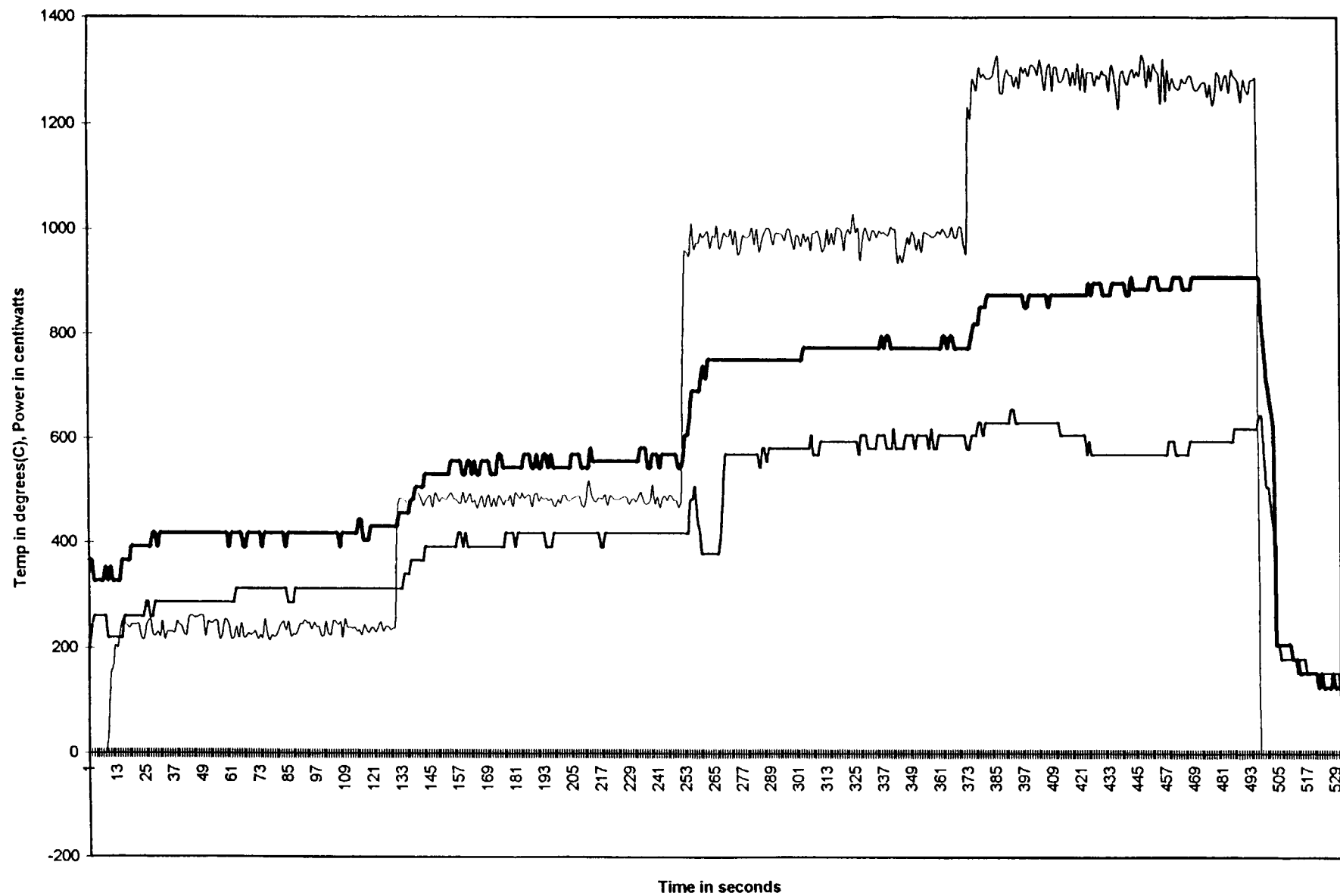
5. References

1. Purves, L. "Robot Operated Materials Processing System (ROMPS), *Critical Design Review (CDR)*," NASA/Goddard Space Flight Center, 1992.
2. Nguyen, C.C., "Testing of ROMPS Robot Mechanical Interfaces and Compliant Device," *Semiannual Progress Report*, NASA/Goddard Space Flight Center, Grant NAG 5-1415, March 1993.
3. Goddard Space Flight Center/NASA , "Robot Operated Materials Processing System (ROMPS), Phase III, Flight Safety Data Package," *GSFC/NASA*, April 1994.

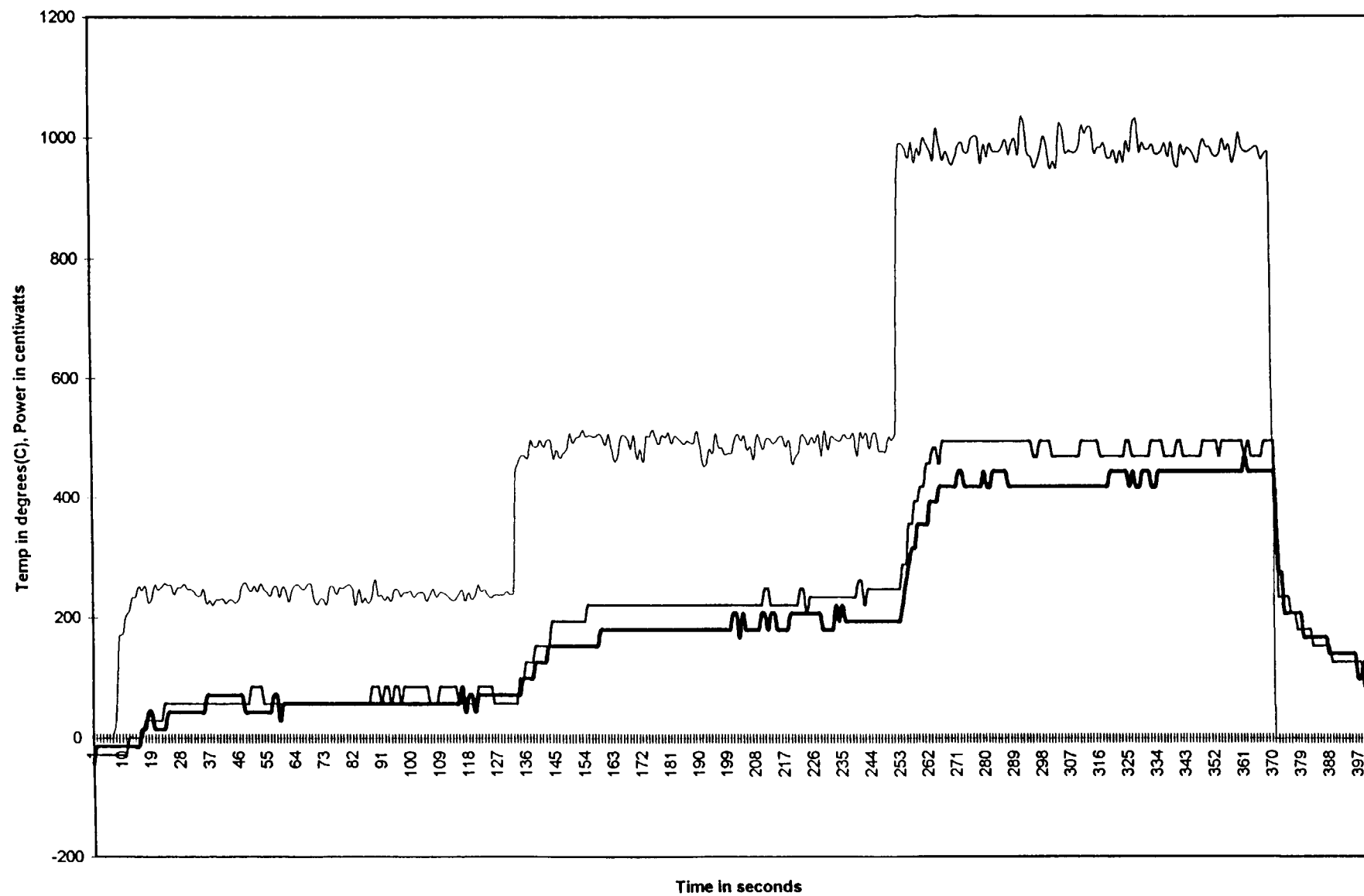
Appendix 1

Data Processing Using MicroSoft Excel

"Processed" Data Only

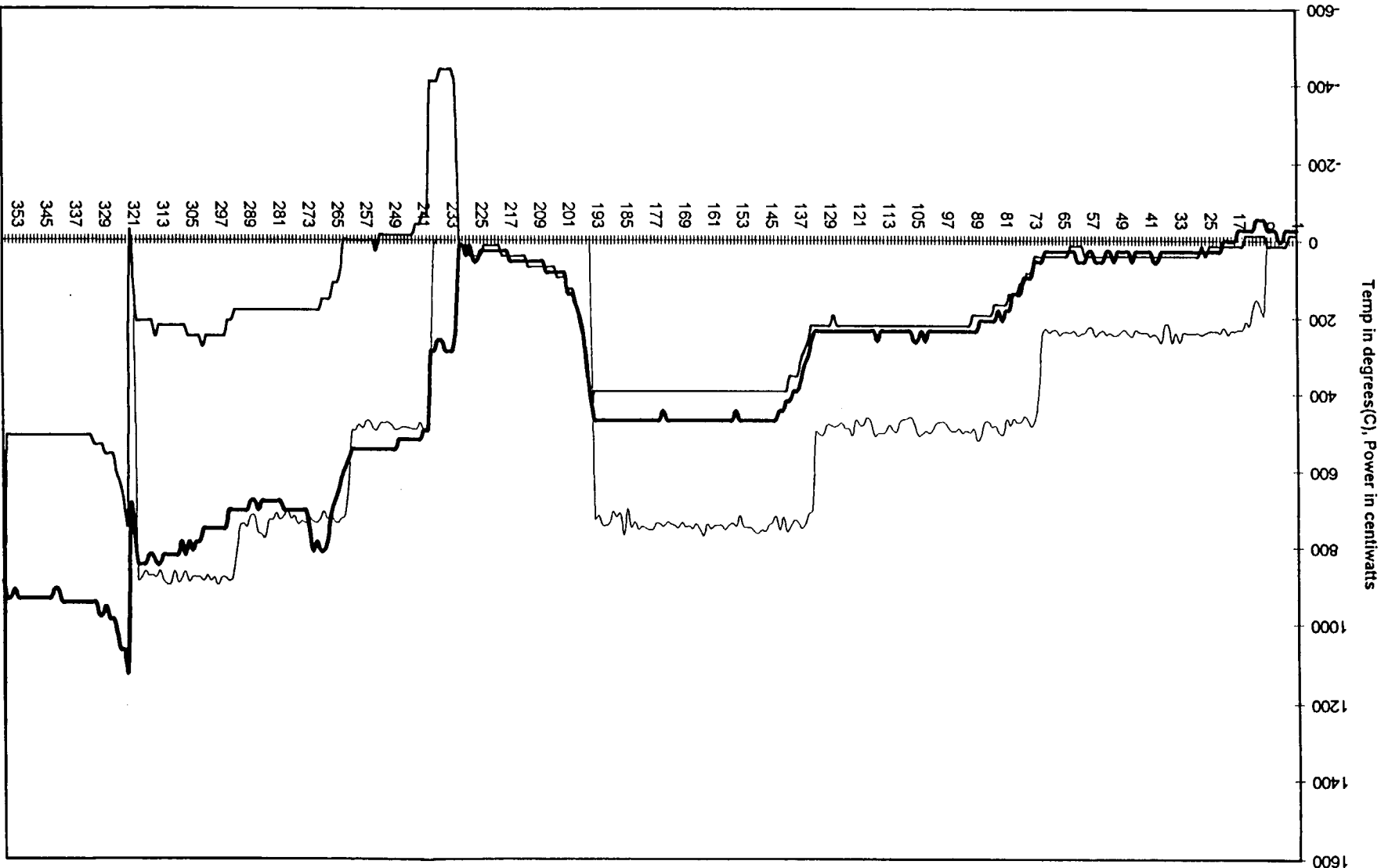


thin line=power; normal line=temp C1; thick line=temp C2

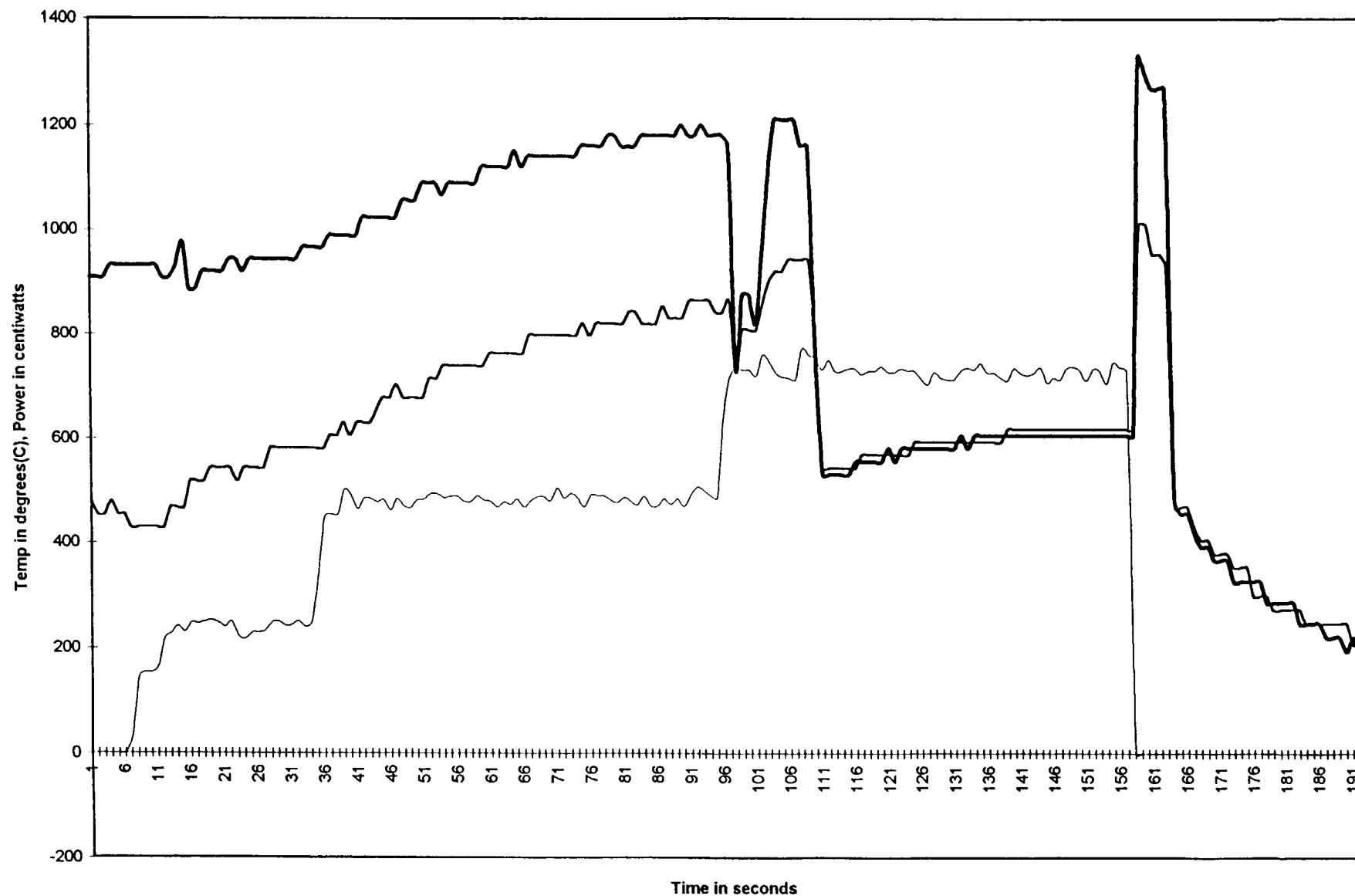


thin line=power; normal line=temp C1; thick line=temp C2

Flight Data, Sample 62, First Run, new62-1.xls

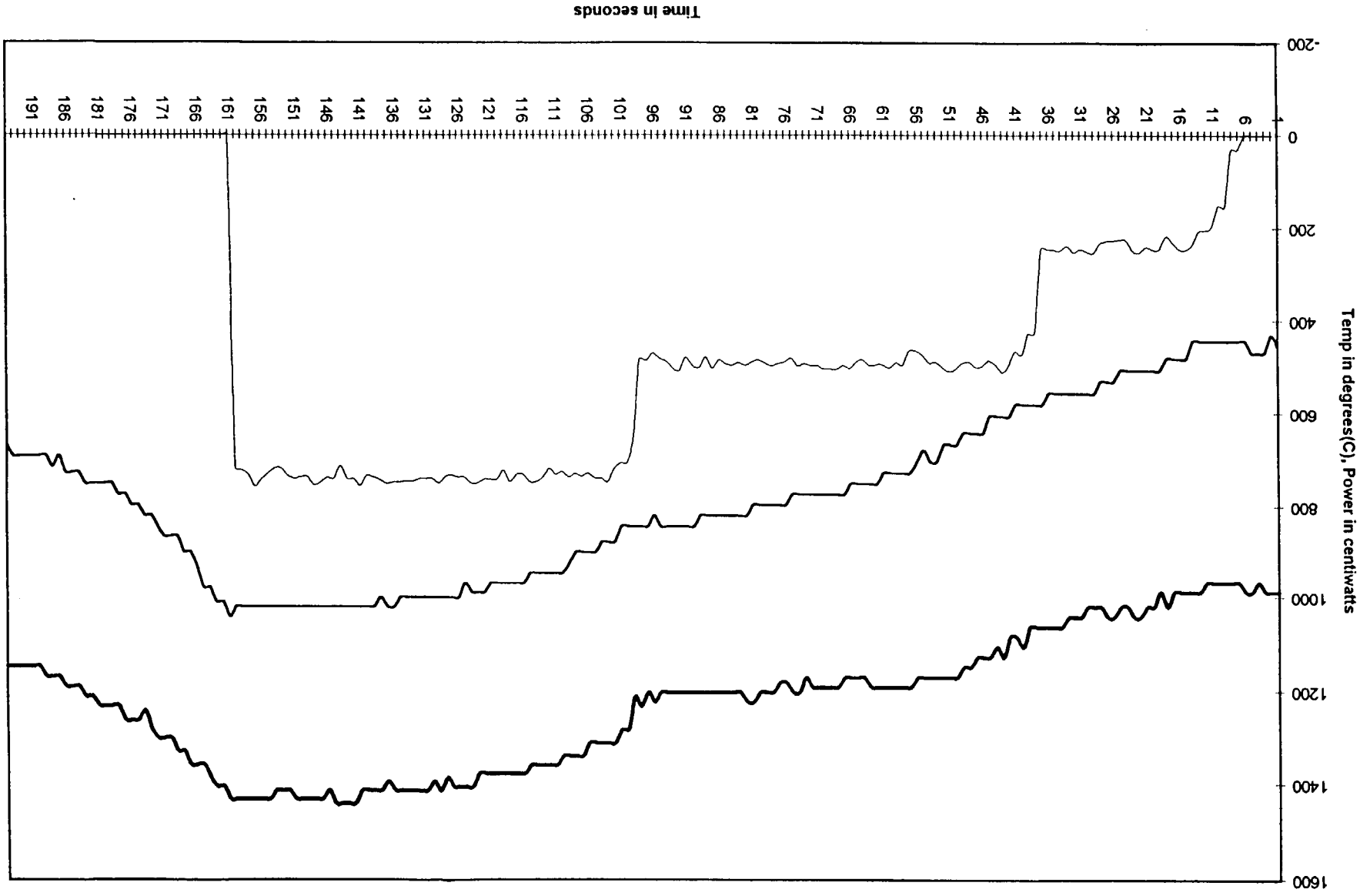


thin line=power; normal line=temp C1; thick line=temp C2



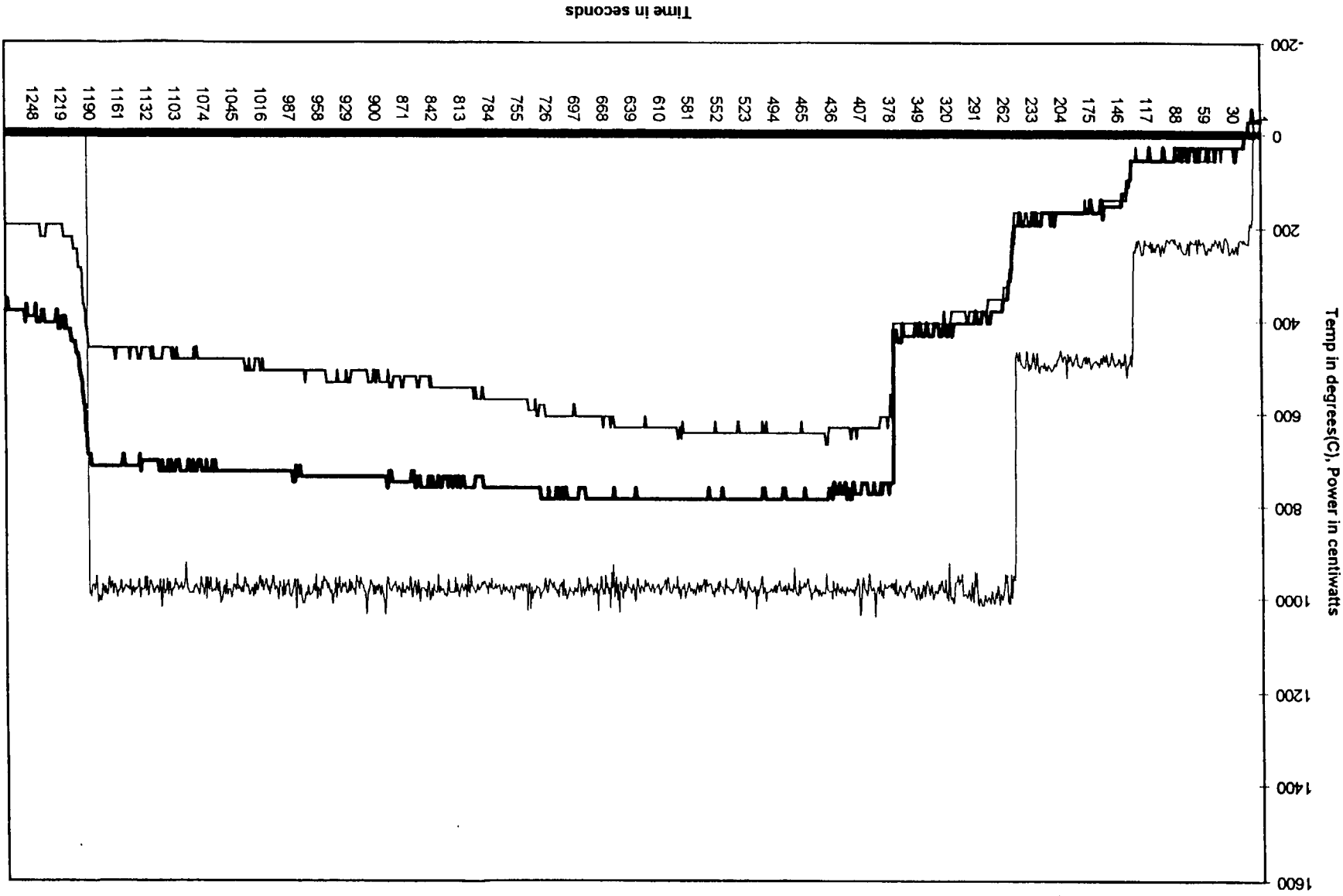
thin line=power; normal line=temp C1; thick line=temp C2

Flight Data, Sample 74, First Run, new74-1.xls



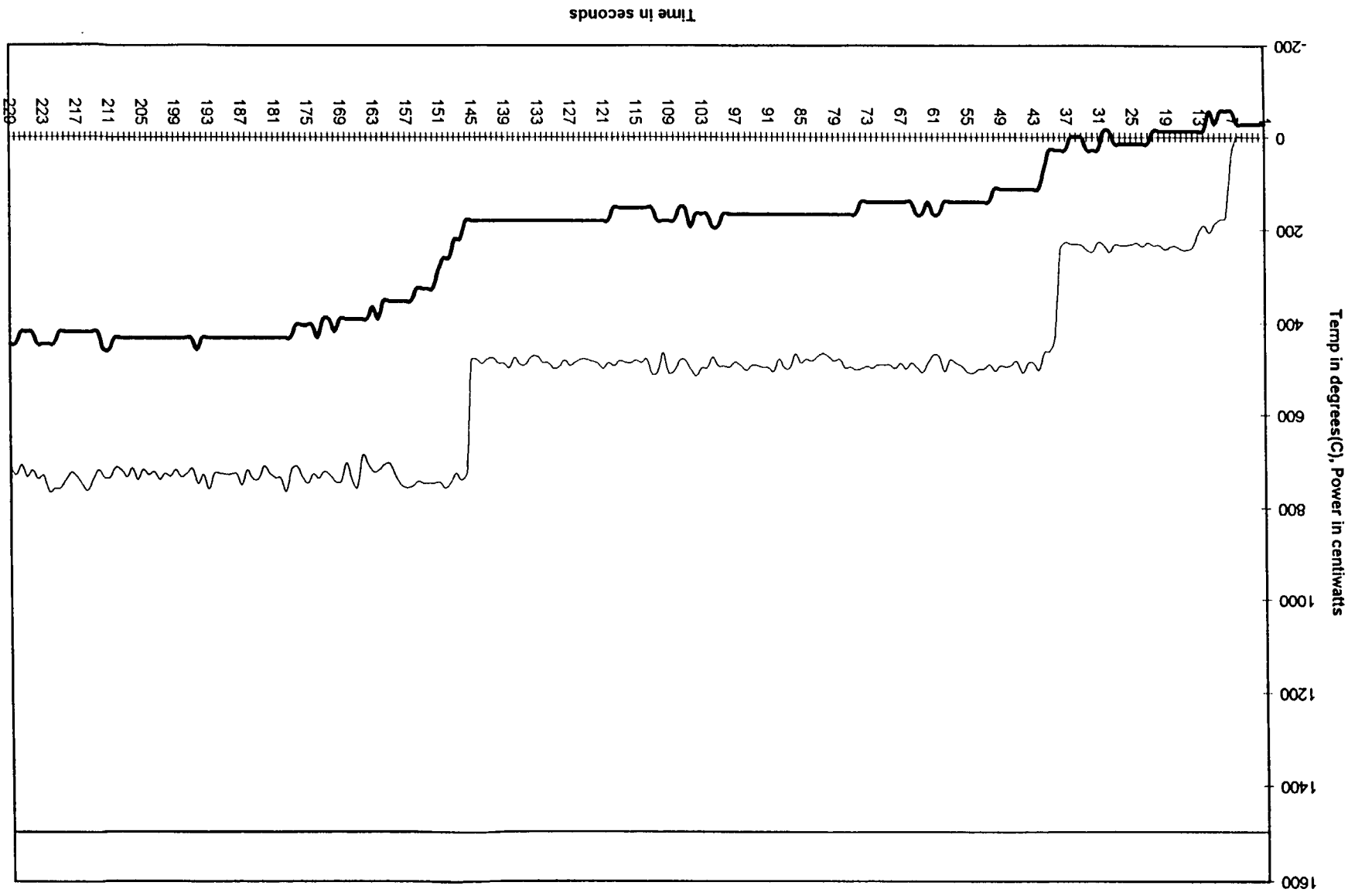
thin line=power, normal line=temp C1, thick line=temp C2

Flight Data, Sample 87, First Run, new87-1.xls



Flight Data, Sample 100, First Run, new100-1.xls

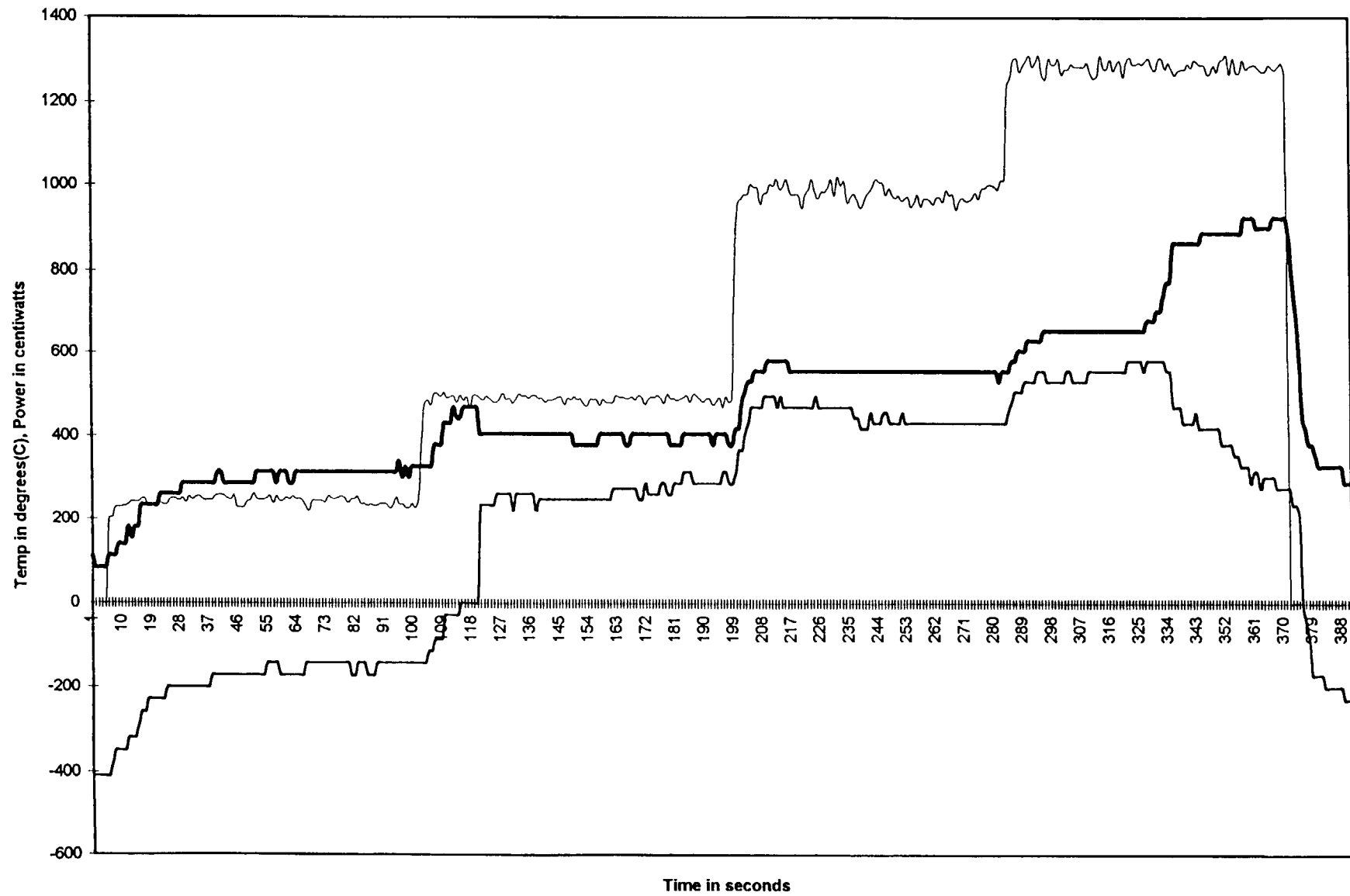
FURNACE B



thin line=power, normal line=temp C1; thick line=temp C2

FURNACE A

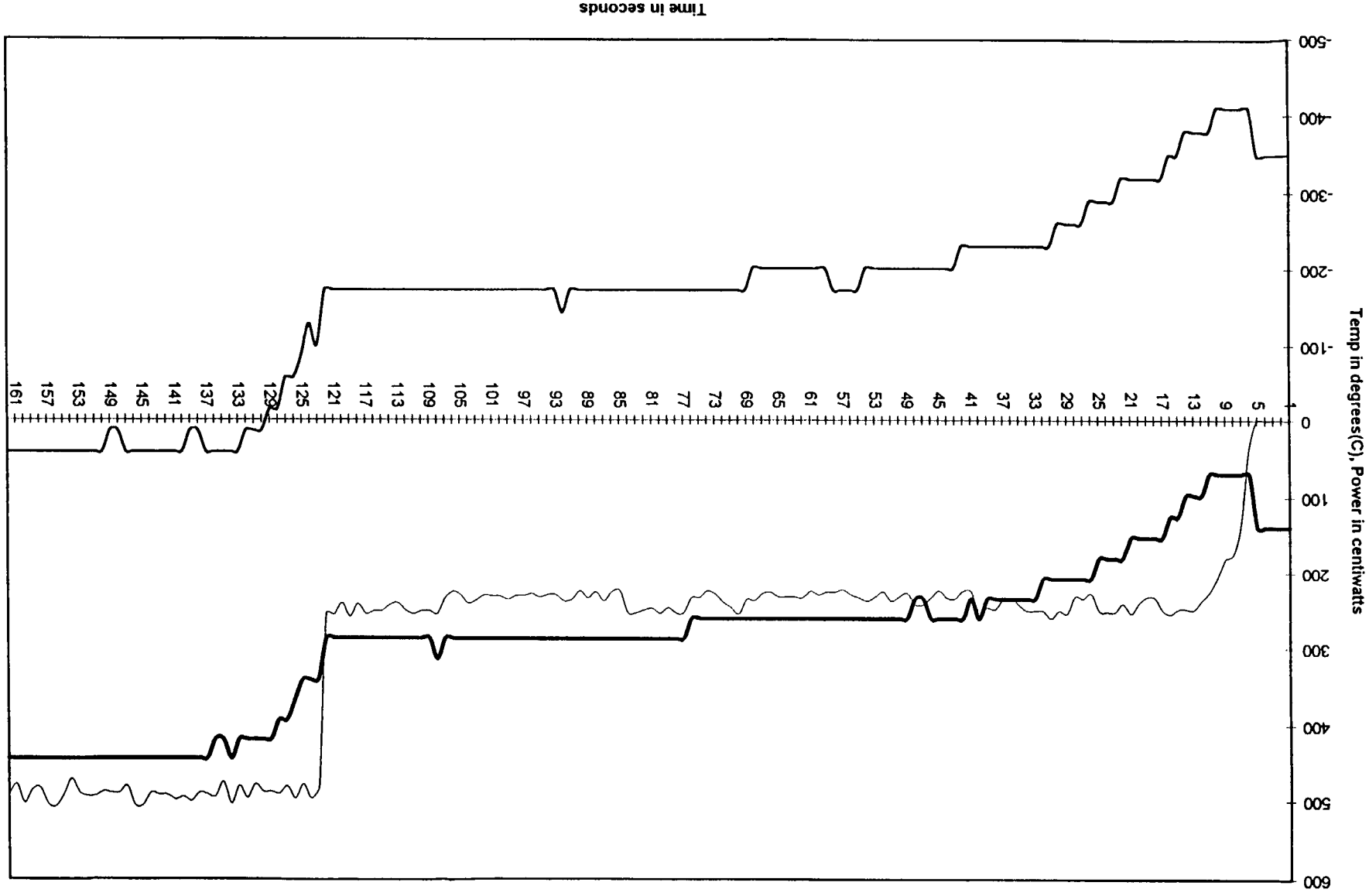
Flight Data, Sample 44, Second Run, new44-2.xls



thin line=power; normal line=temp C1; thick line=temp C2

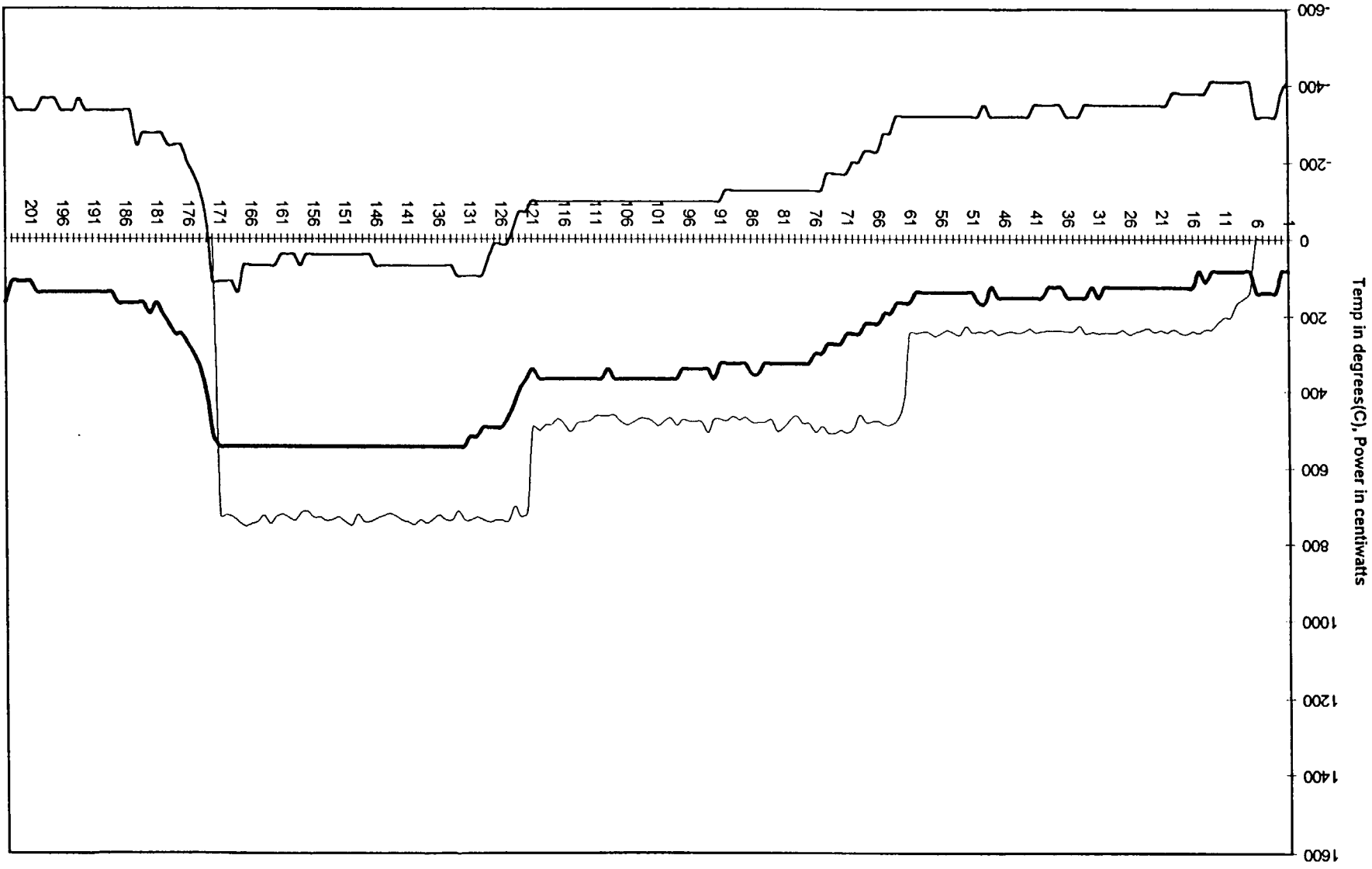
FURNACE A

Flight Data, Sample 45, Second Run, new45-2.xls



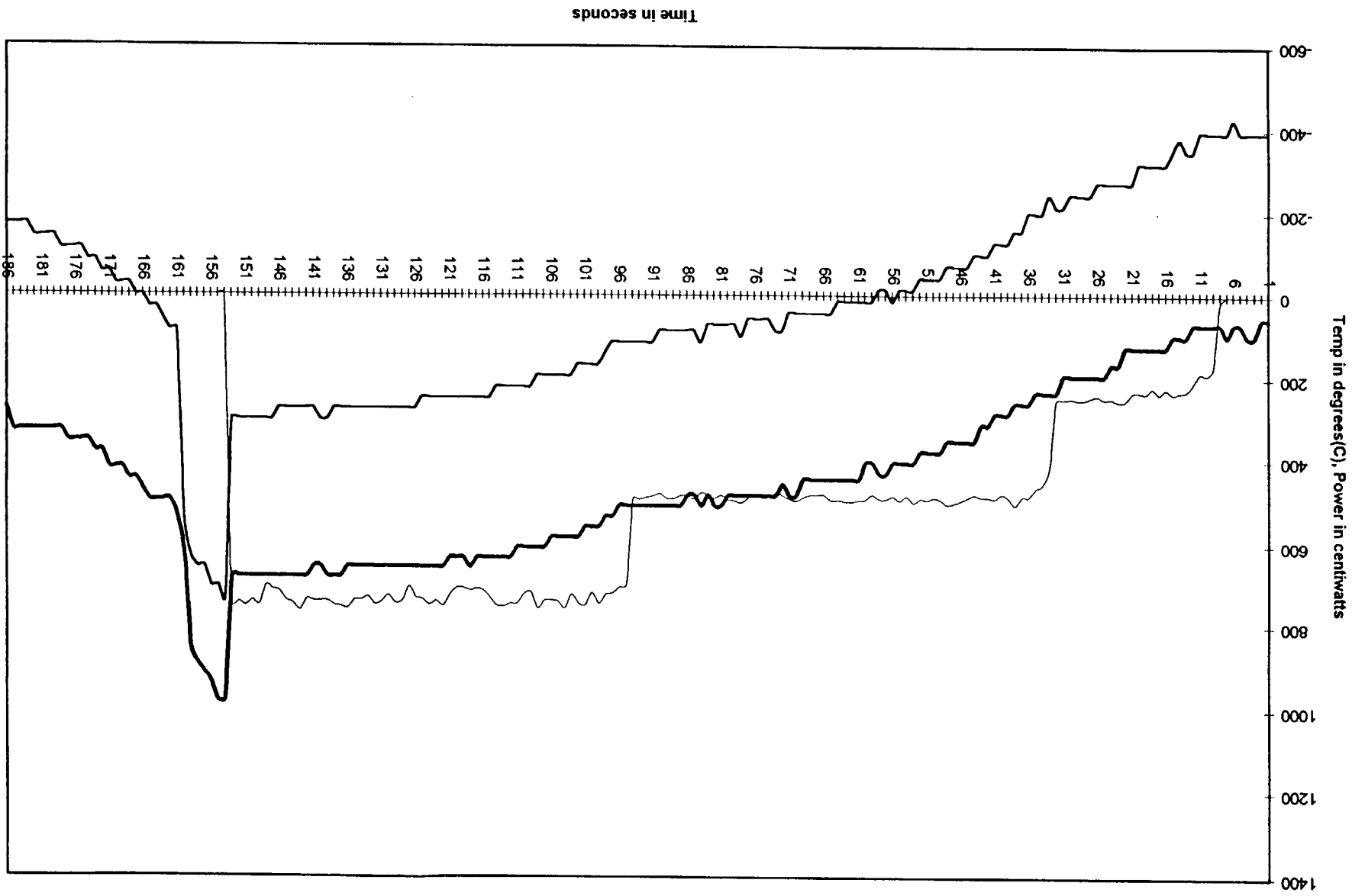
FORNOC #

Flight Data, Sample 62, Second Run, new62-2.xls



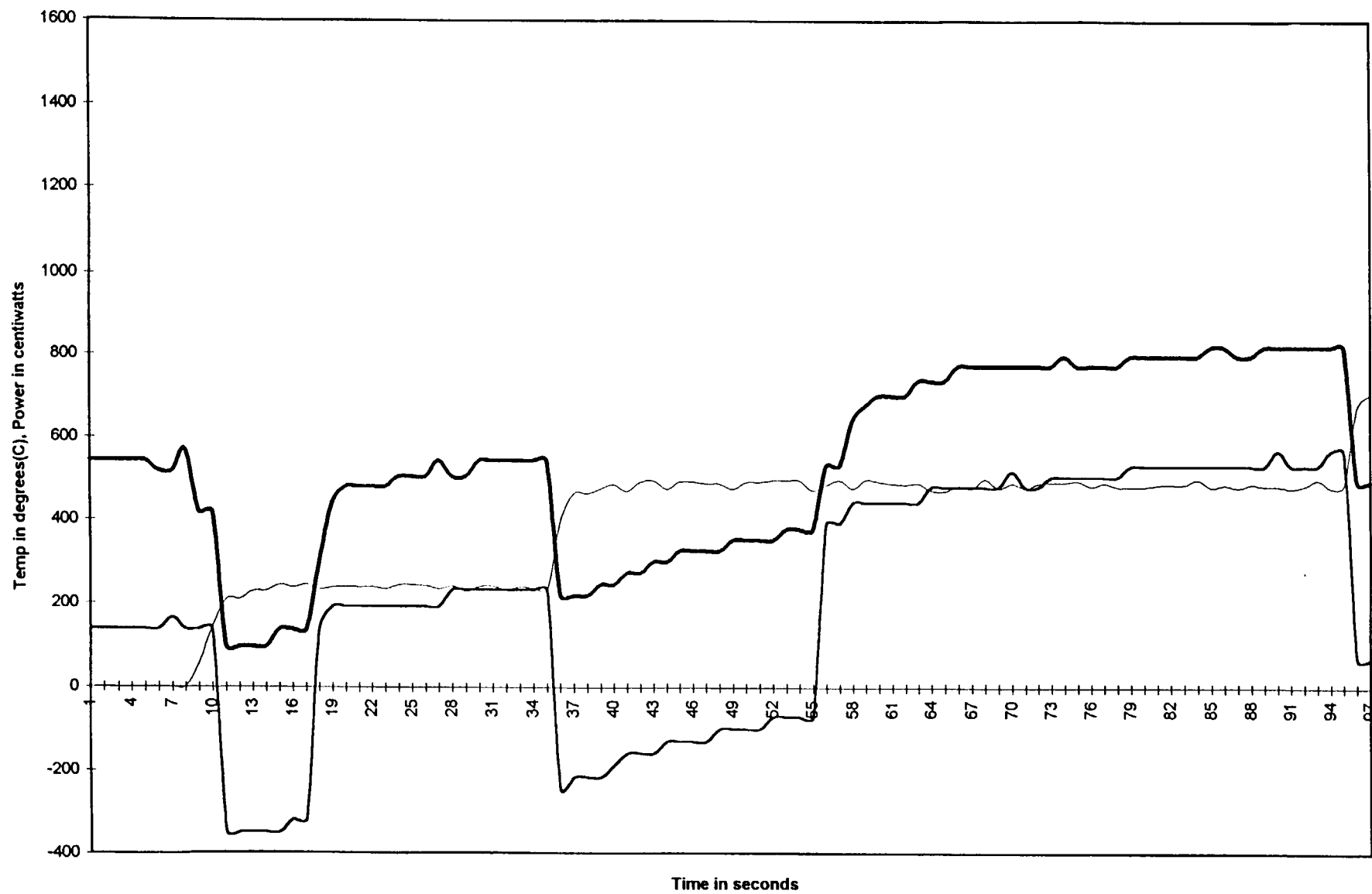
FORNCE A

Flight Data, Sample 73, Second Run, new73-2.xls



FURNACE A

Flight Data, Sample 74, Second Run, new74-2.xls



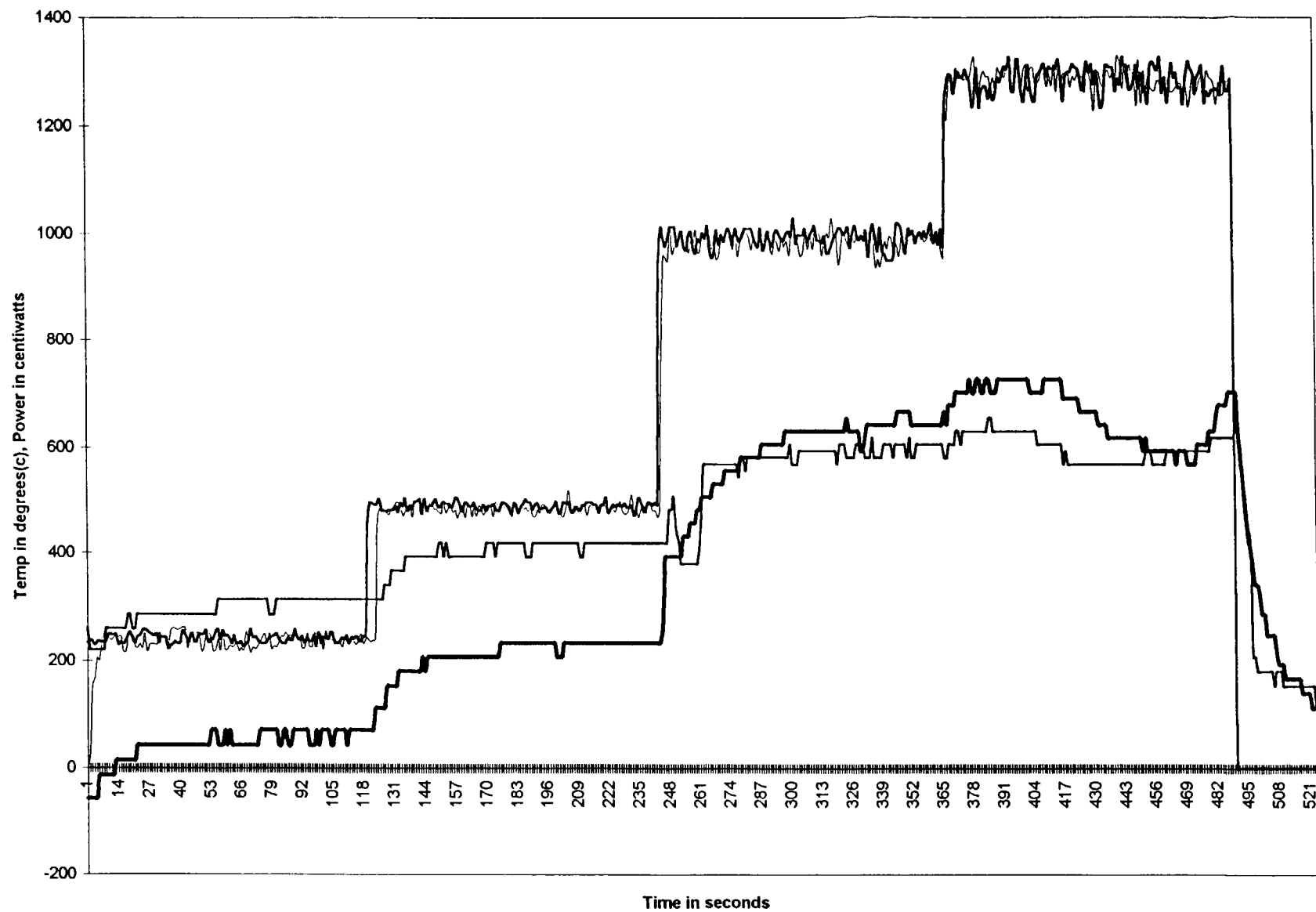
thin line=power; normal line=temp C1; thick line=temp C2

Appendix 2

Data Processing Using MicroSoft Excel

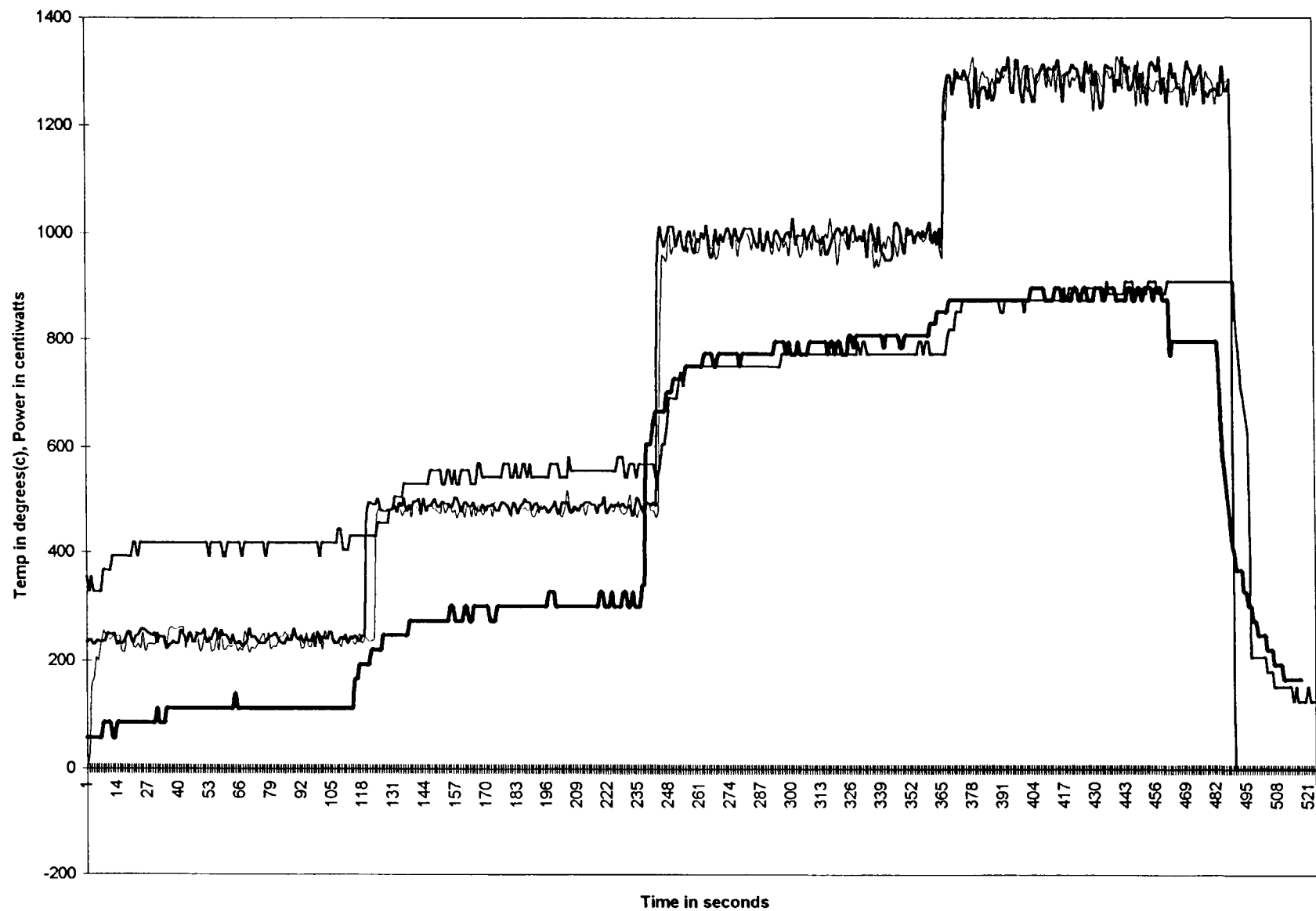
"Processed" Data and Raw Data

Processed Data, Sample 44, Sensor 1, Furnace B, pro44-1.xls



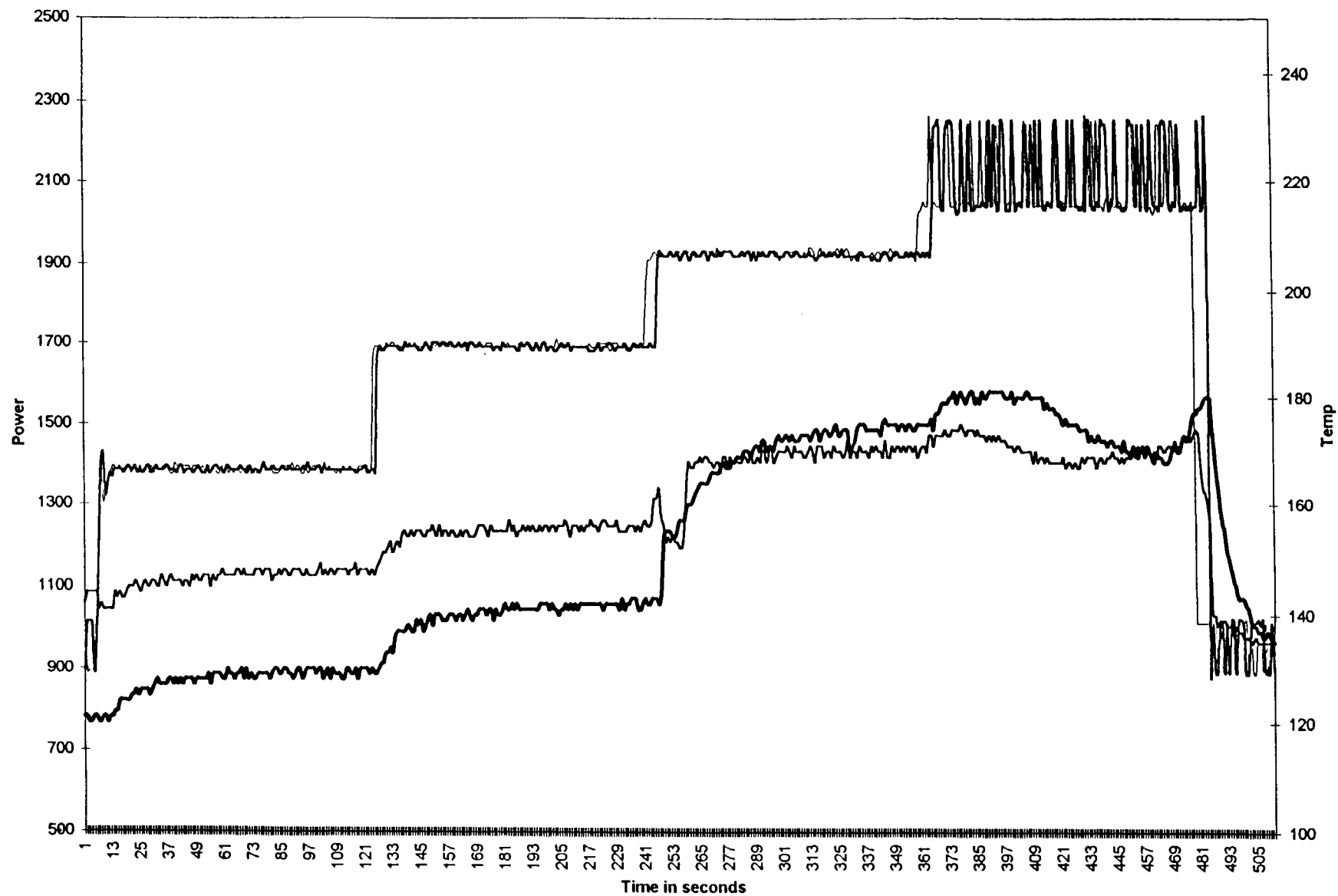
POWER (thin line = flight, normal line = Goddard); DATA (normal line = flight, thick line = Goddard)

Processed Data, Sample 44, Sensor 2, Furnace B, pro44-2.xls

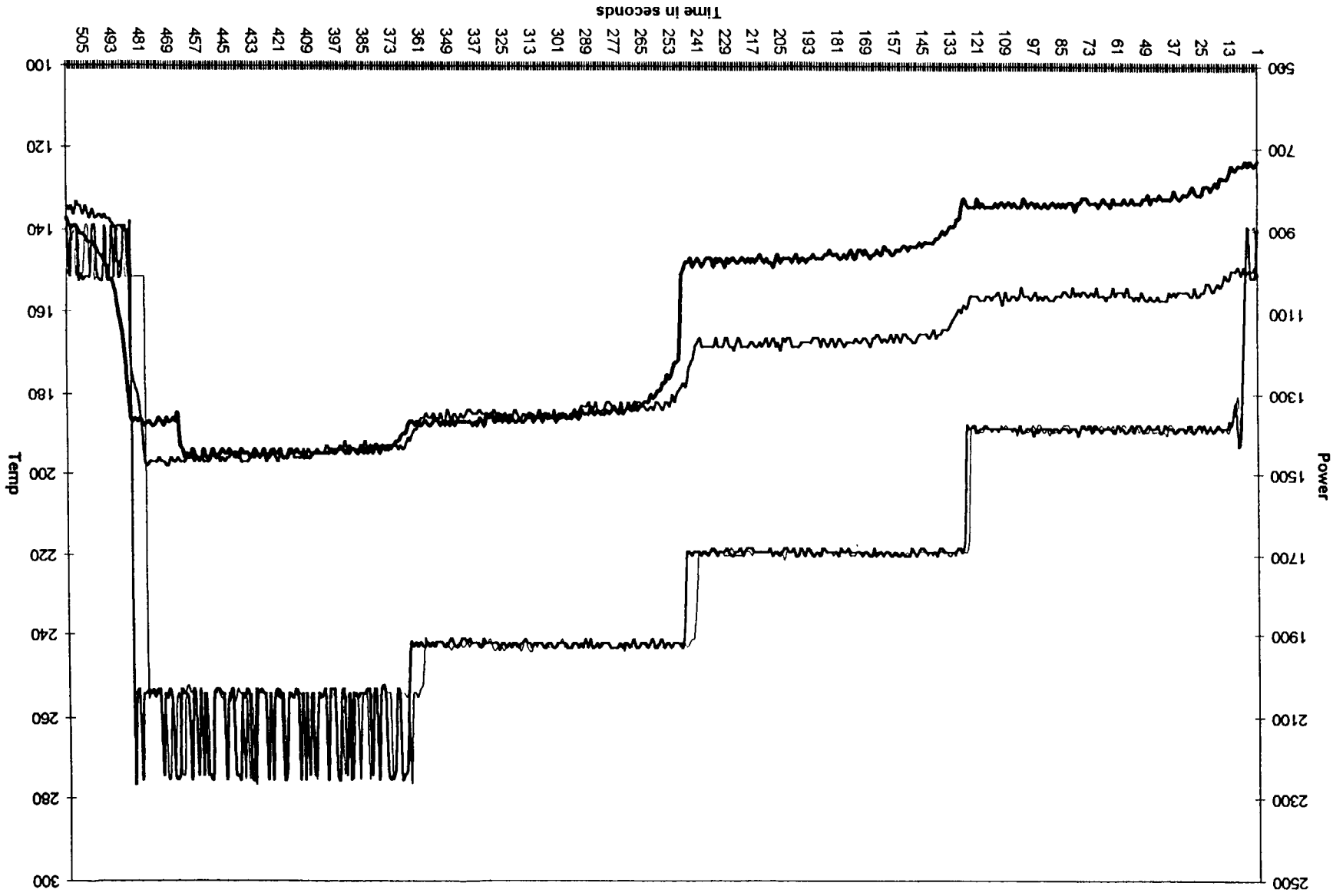


POWER (thin line = flight, normal line = Goddard); DATA (normal line = flight, thick line = Goddard)

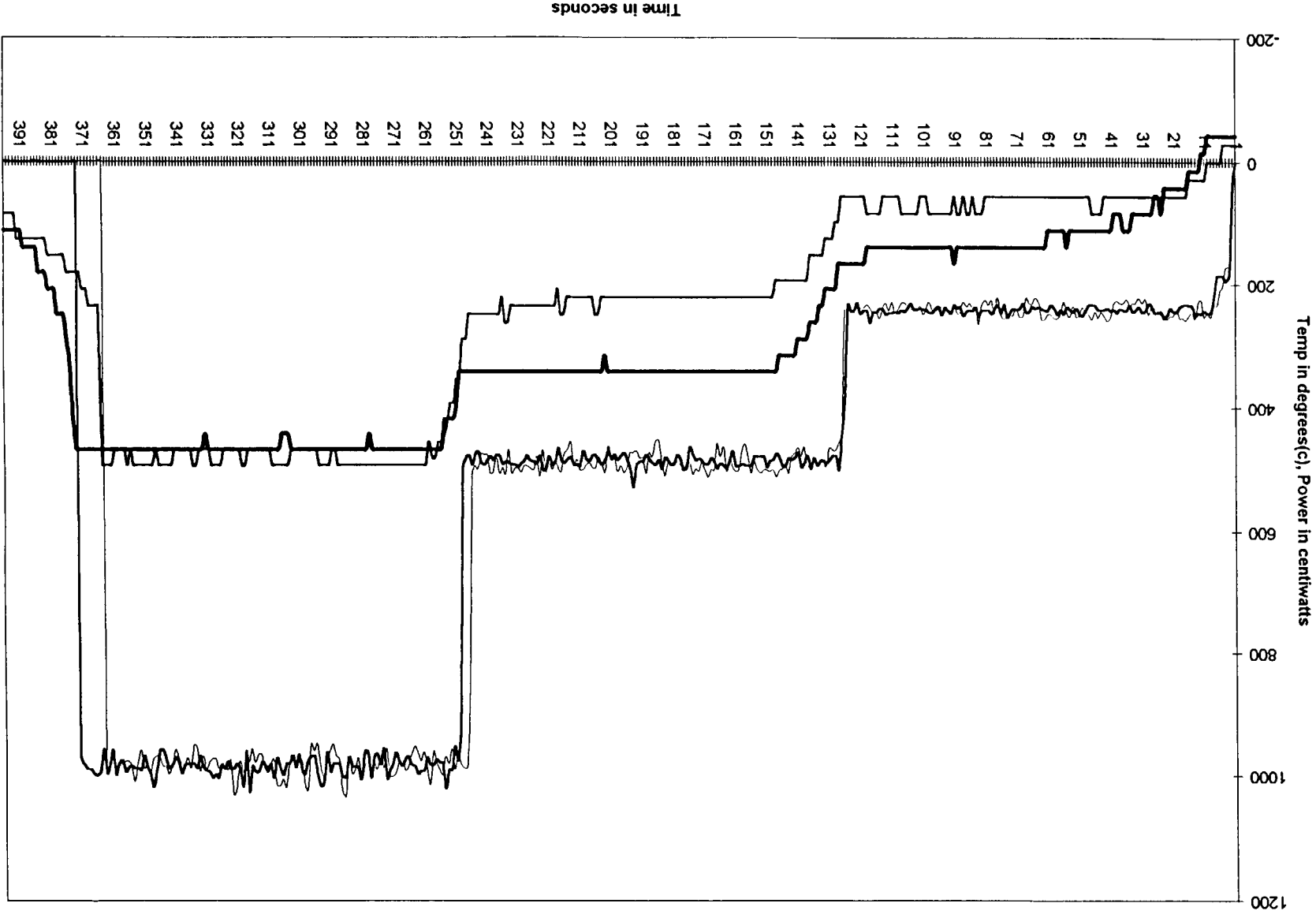
Raw Data, Sample 44, Sensor1, Furnace B, samp44-1.xls



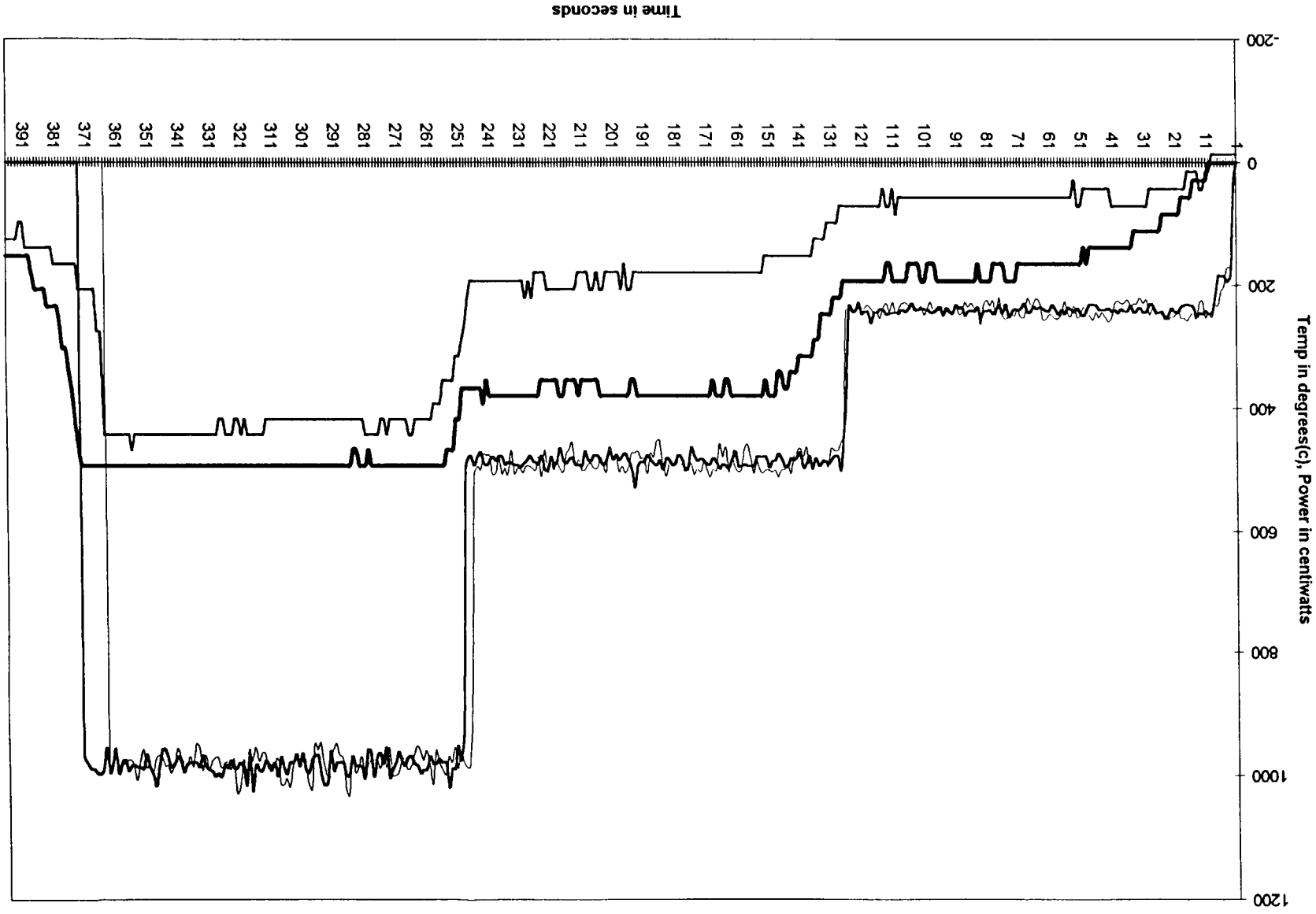
POWER (thin line = flight, normal line = Goddard); DATA (normal line = flight, thick line = Goddard)



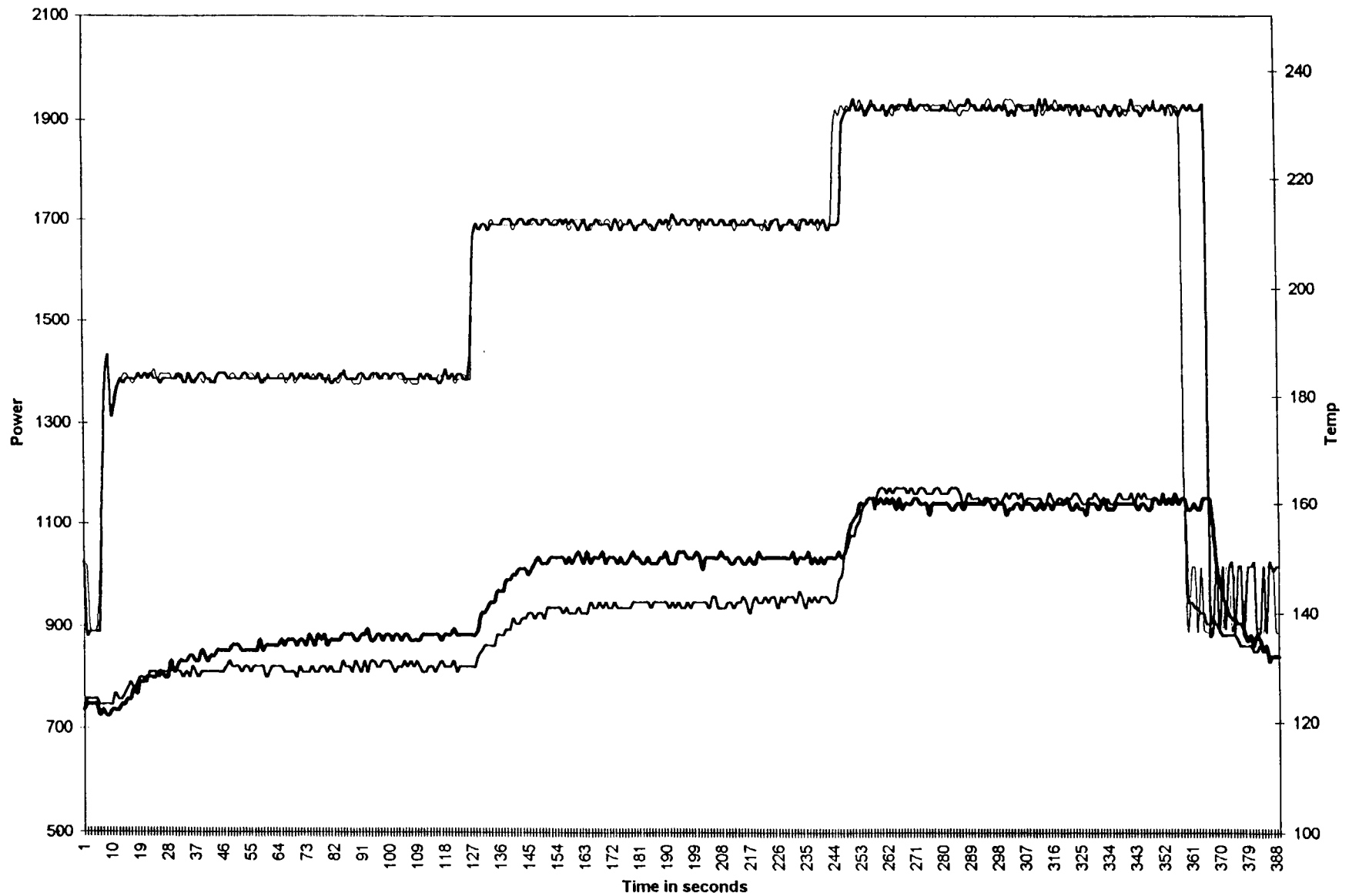
POWER (thin line = flight, normal line = Goddard); DATA (normal line = flight, thick line = Goddard)



POWER (thin line = flight, normal line = Goddard); DATA (normal line = flight, thick line = Goddard)

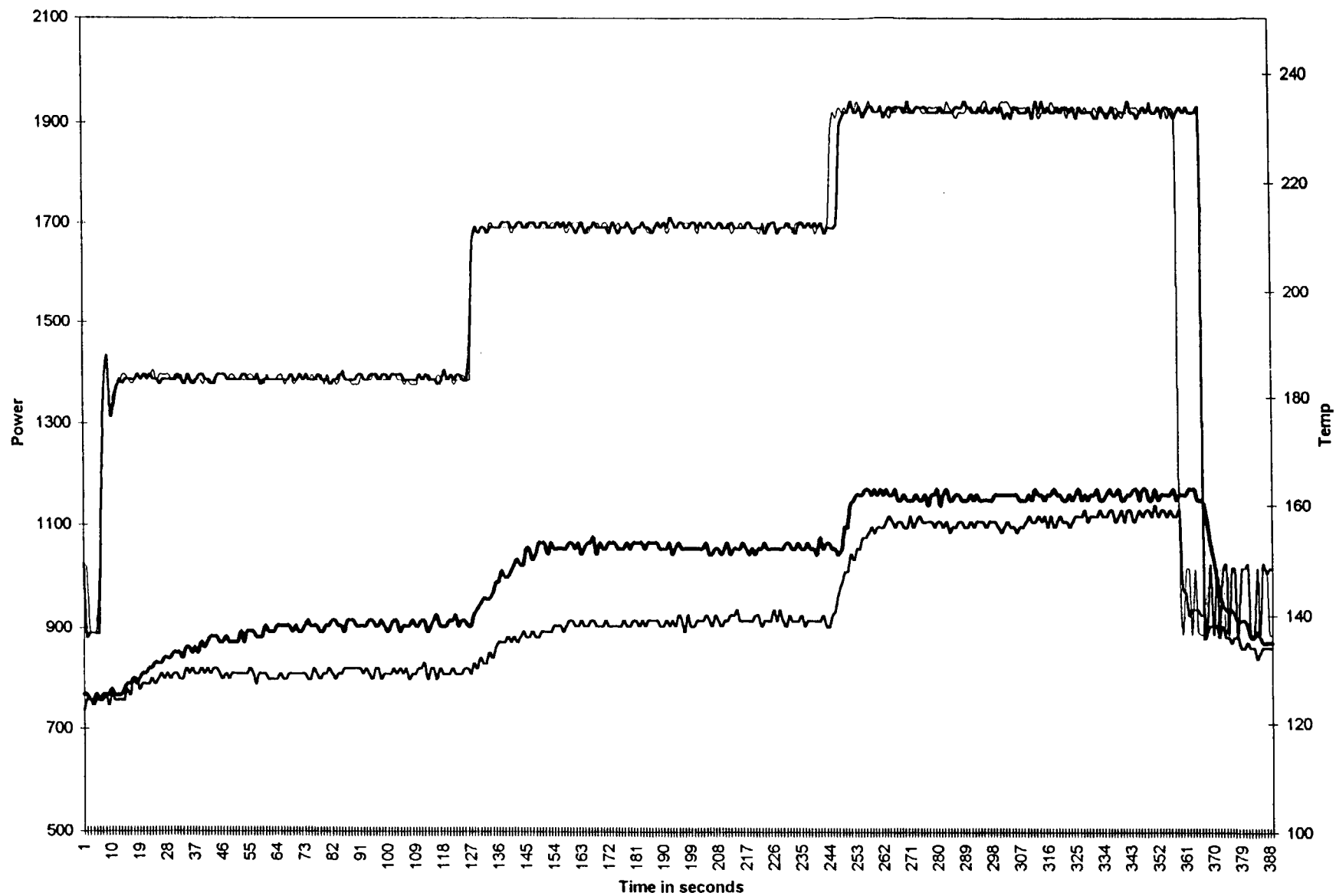


Raw Data, Sample 45, Sensor1, Furnace B, samp45-1.xls

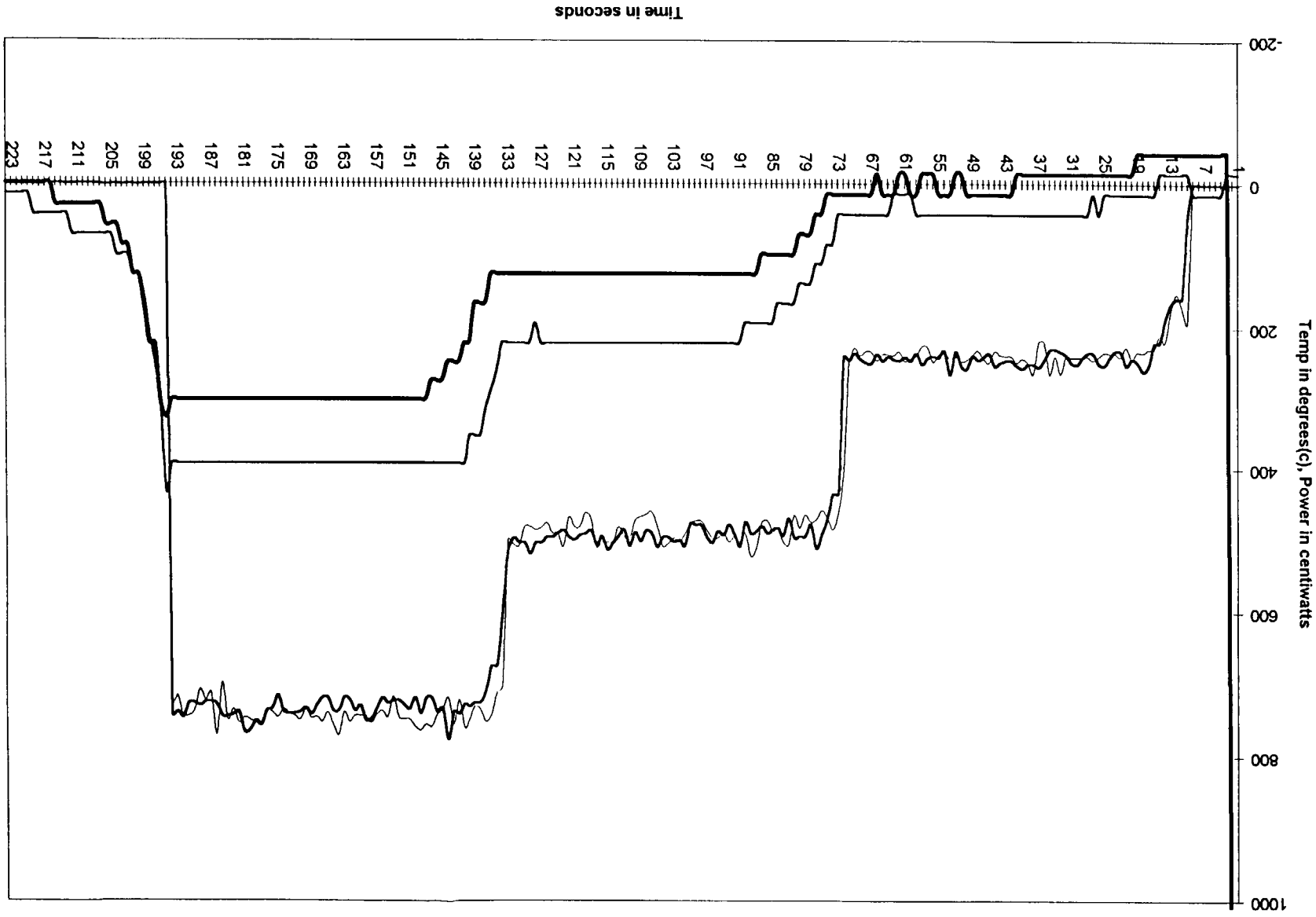


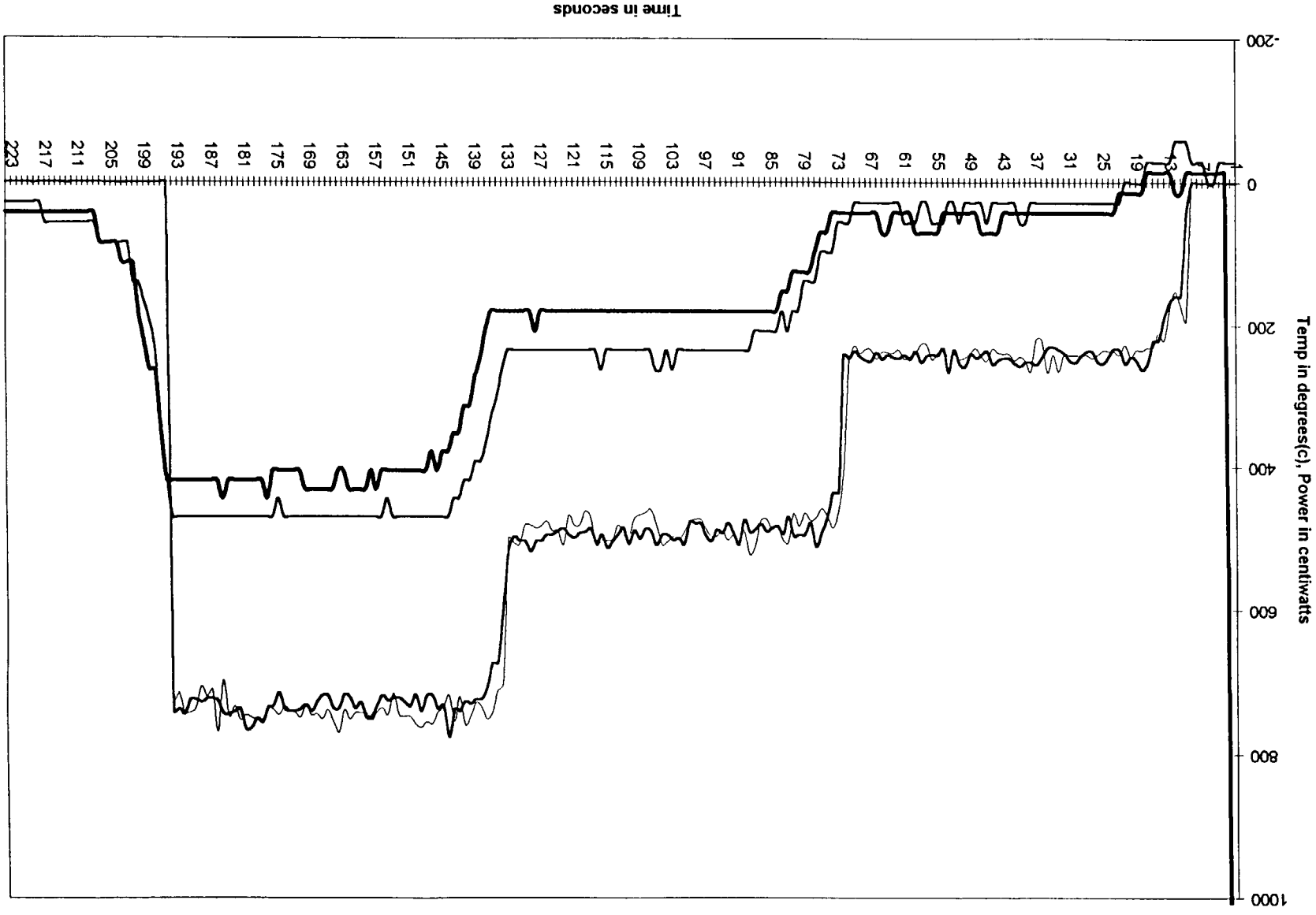
POWER (thin line = flight, normal line = Goddard); DATA (normal line = flight, thick line = Goddard)

Raw Data, Sample 45, Sensor2, Furnace B, samp45-2.xls

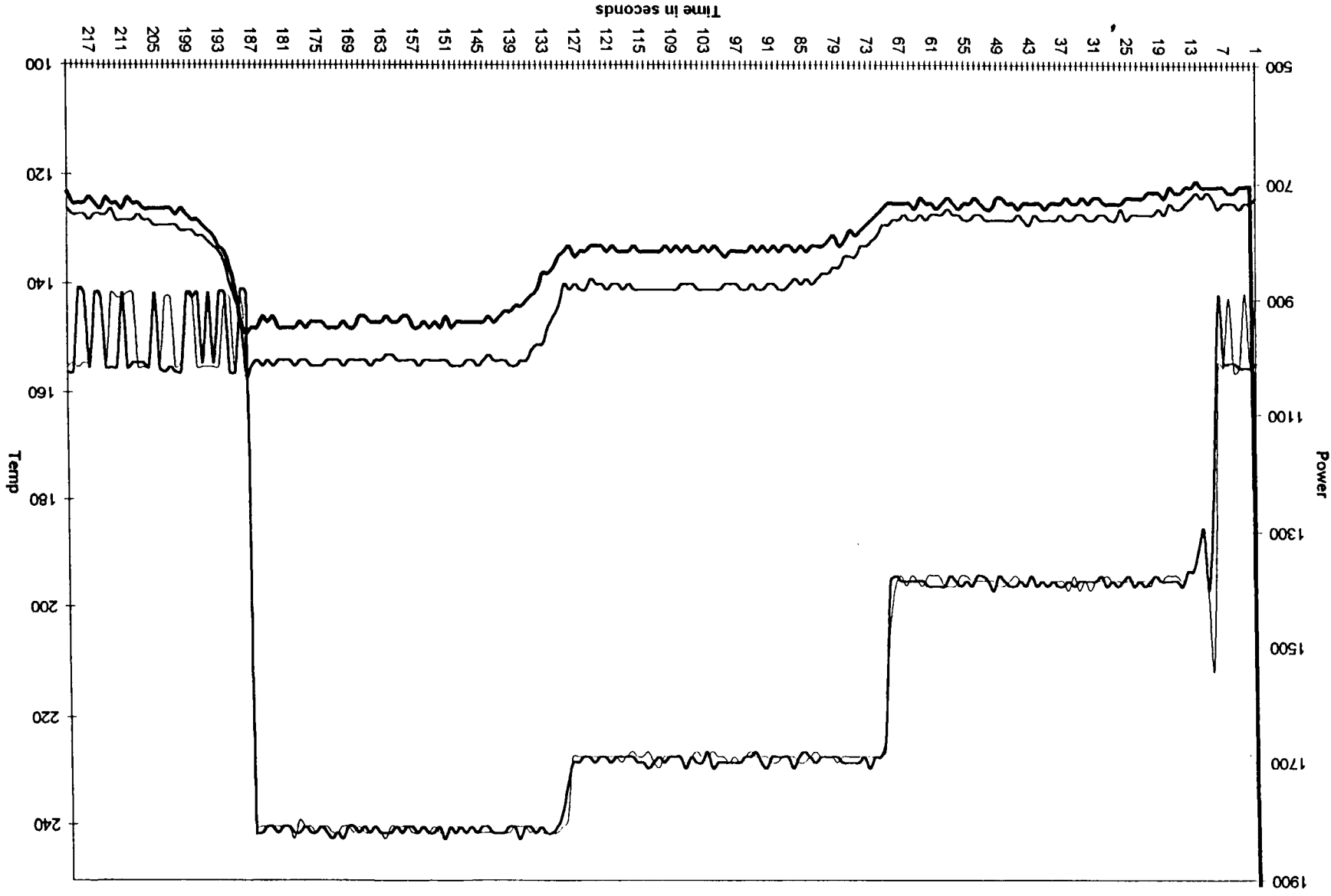


POWER (thin line = flight, normal line = Goddard); DATA (normal line = flight, thick line = Goddard)



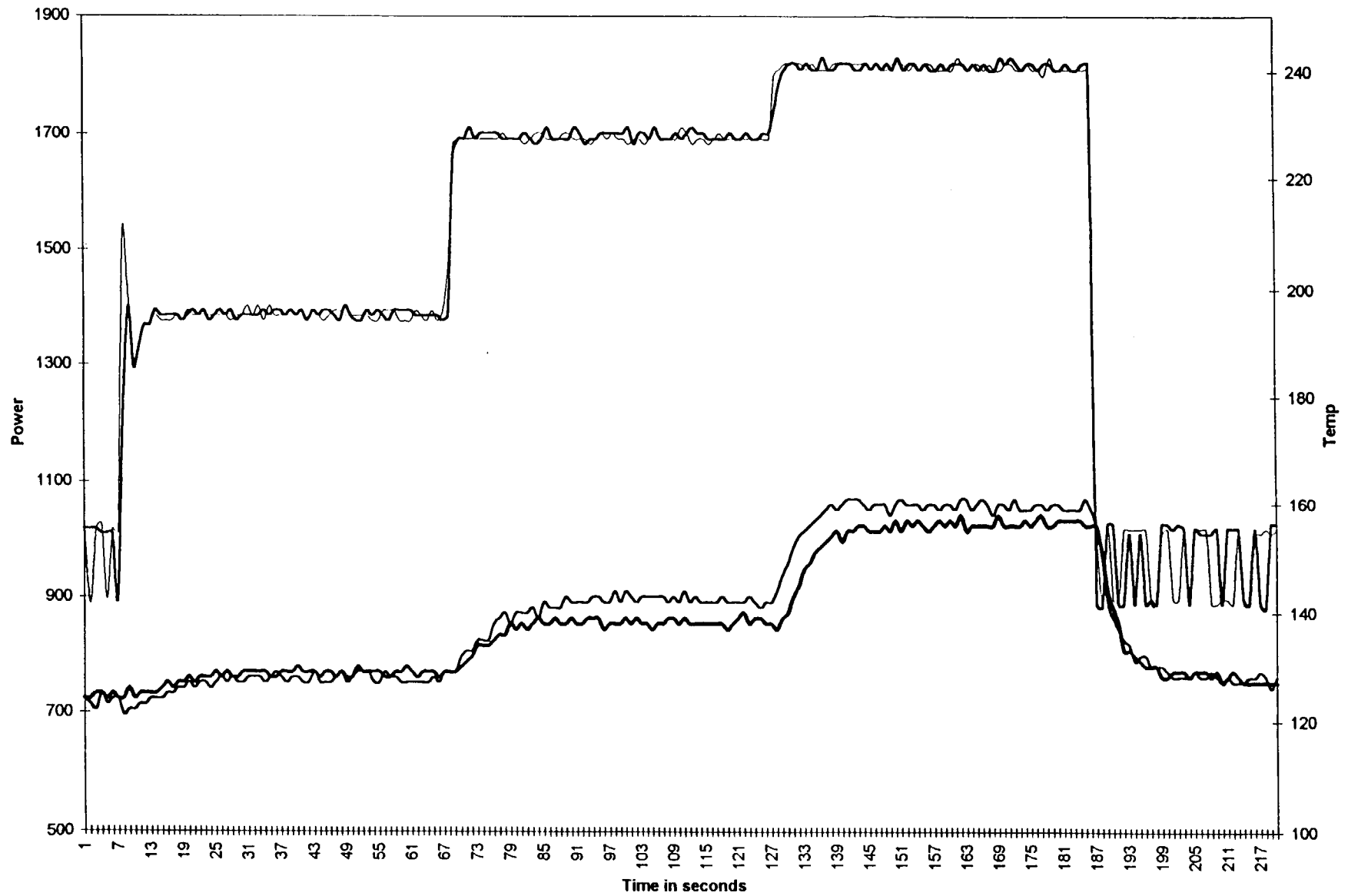


POWER (thin line = flight, normal line = Goddard); DATA (normal line = flight, thick line = Goddard)



POWER (thin line = flight, normal line = Goddard); DATA (normal line = flight, thick line = Goddard)

Raw Data, Sample 62, Sensor2, Furnace B, samp62-2.xls



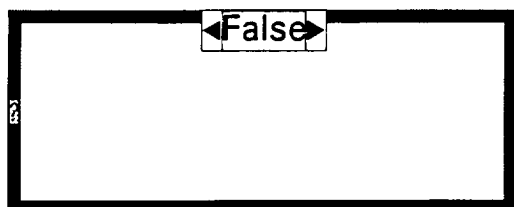
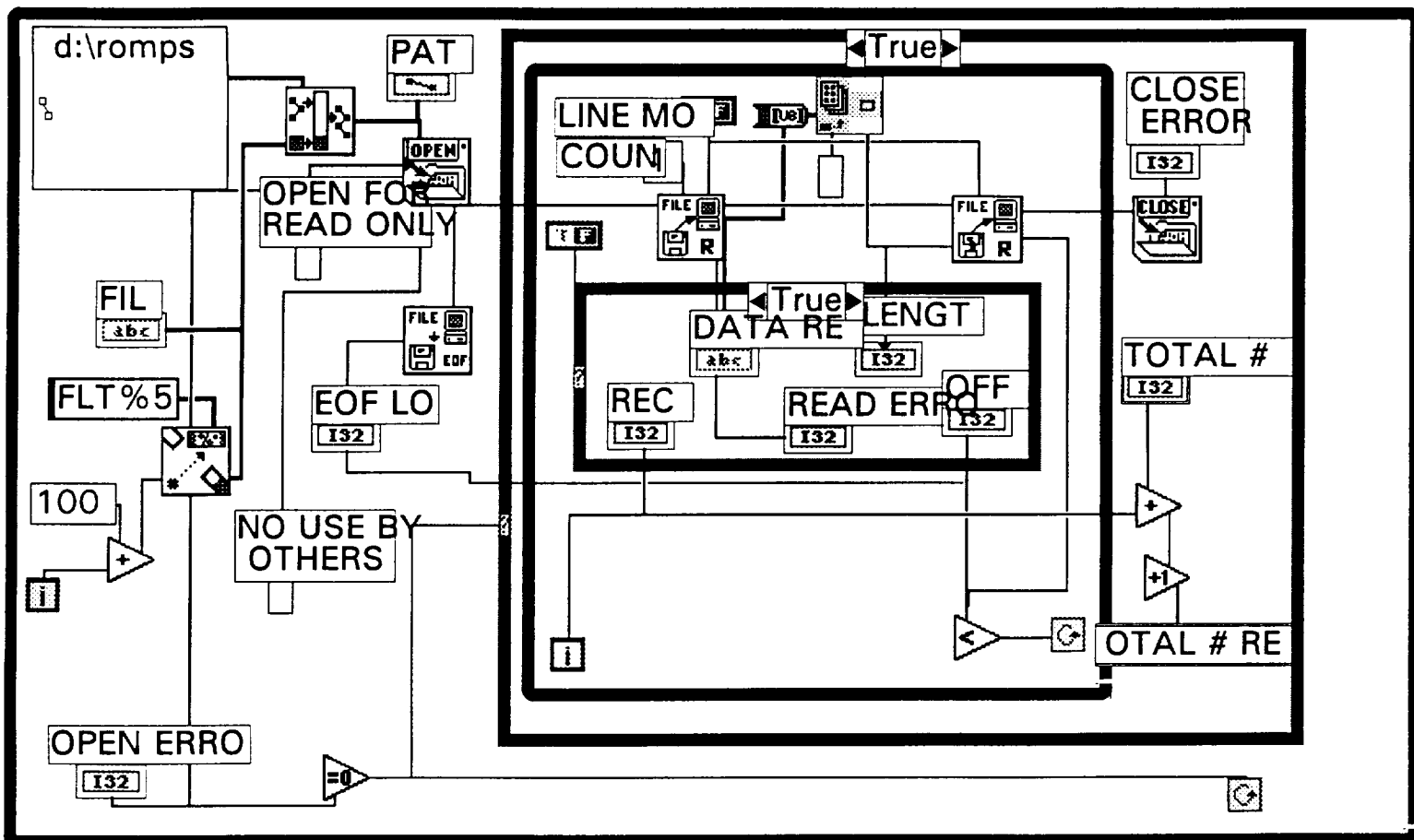
POWER (thin line = flight, normal line = Goddard); DATA (normal line = flight, thick line = Goddard)

Appendix 3

Data Processing Using LabView

Diagram and Front Panel

IOTEST3
10/02/95 02:01 PM



IOTEST3
10/02/95 02:01 PM

DATA RE				REC	0
		READ ERRO	0	OFFSE	LENGT
				0	0
OPEN ERRO	EOF LO				
0	0				
FIL		TOTAL # R	0	CLOSE	
				ERROR	
				0	
PAT					

Appendix 4

Data Processing Using C Program

Listing of Source Codes

/******

This program use screen graphics only with menu

This program is written by Catholic University to process data obtained from ROMPS Flight. In particular it will read the furnace voltage, furnace current, temperature from sensors 1 and 2 of cal samples. Identification Number (ID#) is assigned to cal samples to facilitate programming. A list of the cal sample ID is given below:

Cal Sample	ID#
44 (First Run)	101
44 (Second Run)	102
45 (First Run)	103
45 (Second Run)	104
62 (First Run)	105
62 (second Run)	106
73 (First Run)	107
73 (Second Run)	108
74 (First Run)	109
74 (Second Run)	110
87 (First Run)	111
87 (second Run)	112
100 (First Run)	113
100 (Second Run)	114

*****/

```
#include <stdio.h>
#include "dos.h"
#include <stdlib.h>
#include <bios.h>
```

```

#include <gbuild.h>
#include <graphics.h>
#include "d:\gb\examples\MyPath.h"
char ch1, fn[9],ss[80], sn[40], out;
long int i, nfile, nrec, nchr, nrecc,fvolt, fcurr,power,samp;
long int tstamp, tstart, tend, fstart, fend, typrec, subtyp, tcal1, tcal2;
unsigned char tlmbk[120];
FILE *fp = NULL, *fpo=NULL;
long int nsamp, npoints;
long int tstart_a[14]={37452, 59498, 38174, 69304, 34391, 55701, 32463,
                      63965, 32886, 64362, 39048, 57132, 53865, 55141};
long int tend_a[14] = {37934, 59978, 38541, 69667, 34574, 55883, 32616,
                      64118, 33078, 64515, 40219, 57495, 54142, 55413};
long int fstart_a[14]={28, 49, 28, 59, 25, 45, 23, 54, 23, 54, 29, 47, 43, 45};
long int fend_a[14]={30, 51, 31, 62, 27, 47, 25, 56, 25, 57, 32, 49, 46, 47};
int sampstart,sampend,samp1,samp2;

/* Chart Variables */
static char    Title[]    = "NASA ROMPS Flight Data";
static char    SubTitle[] = "Sample 1";
static char    TimeTxt[]  = "Time in seconds";
static double  Time[500];
static char    PowerTxt[] = "Power / 10";
static double  Power[500];
static char    Temp1Txt[] = "Temperature 1";
static double  Temp1[500];
static char    Temp2Txt[] = "Temperature 2";
static double  Temp2[500];
static ChartDef ch;
int ierr, drv=DETECT, mode;
/* ***** Change as necessary ***** */
static char *MyPath  = "d:\gd45\bgi"; /* path to BGI files */
static char *MyPrinter = "_LJ3R";    /* driver to use */

```

```

static int MyPort    = PortLPT1B;    /* I/O Port */
static int MyMode    = LandMed;      /* page format. any driver has a 0 */
/* ***** */

```

```

main(){ /* Starting main menu, OPEN CURLY BRACE 1 */
    int printerdrv, printermode;
    unsigned printererr;
    menu:
    printf("The available options are:\n");
    printf("1. Processing a single sample\n");
    printf("2. Processing multiple sample\n");
    printf("3. Cancel\n\n");
    printf("Please select your choice by entering 1 or 2: \n\n\n");
    ch1=getch();
    switch(ch1){/* OPEN CURLY BRACE 2 */
        case '1':
            menu1:
            printf("CAL SAMPLE          ID#\n\n");
            printf("44 (First Run)      1\n");
            printf("44 (Second Run)     2\n");
            printf("45 (First Run)      3\n");
            printf("45 (Second Run)     4\n");
            printf("62 (First Run)      5\n");
            printf("62 (Second Run)     6\n");
            printf("73 (First Run)      7\n");
            printf("73 (Second Run)     8\n");
            printf("74 (First Run)      9\n");
            printf("74 (Second Run)    10\n");
            printf("87 (First Run)     11\n");
            printf("87 (Second Run)    12\n");
            printf("100 (First Run)    13\n");

```

```
printf("100 (Second Run)    14\n\n");
printf("PLEASE SELECT YOUR SAMPLE BY ENTERING ITS ID#\n\n\n\n");
```

```
scanf("%d", &samp);
if (samp<=14) {
    sampstart=samp;
    sampend=samp;
    goto start;}
if (samp>14) {
    printf("YOU HAVE SELECTED AN INVALID SAMPLE\n");
    printf("PLEASE RETRY\n");
    goto menu1;}
```

```
case '2':
```

```
menu2:
```

```
printf("CAL SAMPLE        ID#\n\n");
printf("44 (First Run)    1\n");
printf("44 (Second Run)     2\n");
printf("45 (First Run)        3\n");
printf("45 (Second Run)       4\n");
printf("62 (First Run)         5\n");
printf("62 (Second Run)       6\n");
printf("73 (First Run)         7\n");
printf("73 (Second Run)       8\n");
printf("74 (First Run)         9\n");
printf("74 (Second Run)      10\n");
printf("87 (First Run)        11\n");
printf("87 (Second Run)      12\n");
printf("100 (First Run)       13\n");
printf("100 (Second Run)     14\n\n");
```

```
printf("Please enter the starting sample: \n");
scanf("%d", &samp1);
```

```

printf("Please enter the ending sample: \n");
scanf("%d", &samp2);
if (samp1>14 || samp2>14 || samp1>samp2)
{printf("INVALID SAMPLE RANGE SELECTED, PLEASE RETRY\n\n");
goto menu2;}
sampstart=samp1;
sampend=samp2;
goto start;
case '3':
goto end;
default:
printf("You have made an invalid choice\n");
goto menu;
} /* CLOSE CURLY BRACE 2 */
start:
for(nsamp=sampstart;nsamp<=sampend;nsamp++)
{ npoints=0;
tstart=tstart_a[nsamp-1];
tend =tend_a[nsamp-1];
fstart=fstart_a[nsamp-1];
fend =fend_a[nsamp-1];

sprintf(sn,"Processed for Sample %3ld",nsamp+100);
nrecc=0;

fpo =fopen(sn,"w" );
fputs("FILENAME RCNUM TIME FVOLT FCURR CAL1 CAL2 POWER SAMPNUM\n",
fpo);
for (nfile =fstart;nfile<=fend;nfile++)
{sprintf(fn,"FLT%5ld",nfile+10000);
if ((fp=fopen(fn,"rb")) != NULL)
{nrec = 0;

```

```

while (!feof(fp))
{
    fread(tlmbk, 10,1,fp);
    nrec++;nrecc++;
    nchr=tlmbk[9];
    fread( &tlmbk[10],nchr,1,fp);
    nchr+=10;
    typrec=tlmbk[5];
    subtyp=0;
    if (typrec == 1) subtyp=tlmbk[12];
    if (subtyp == 1)
    {
        tstamp = (864000*(tlmbk[14]/16)) + (86400*(tlmbk[14]%16));
        tstamp+= ( 36000*(tlmbk[15]/16)) + ( 3600*(tlmbk[15]%16));
        tstamp+= ( 600*(tlmbk[16]/16)) + ( 60*(tlmbk[16]%16));
        tstamp+= ( 10*(tlmbk[17]/16)) + ( (tlmbk[17]%16));

        if (tstamp >= tstart && tstamp <= tend)
        {
            npoints++;
            fvolt = tlmbk[75];
            fcurr = tlmbk[77];
            Temp1[npoints-1] = tlmbk[78];
            Temp2[npoints-1] = tlmbk[79];
            Power[npoints-1] = fvolt*fcurr/10;
            Time[npoints-1] = tstamp;
        }
    }
    /*sprintf(ss,"%8s%6ld%7ld%7ld%7ld%7ld%10ld%8s\n",fn,nrec,tstamp, fvolt,
    fcurr,tcal1,tcal2,power,sn); */ /* Assign the indicated variables to a string pointed by pointer ss */
    /*fputs(ss,fpo);*/ /* Put the string pointed by ss to the file pointed by fpo (outsamp.dat) */
    /*puts(ss); */ /* Put the string pointed by ss to the screen */
}
}
} /* Go back to while(!feof(fp))*/
fclose(fp);
}

```



```

        /*else goto end;*/ /* When fp=NULL*/
    }; /*Go back to increase nfile by one, see the for loop for (nfile=fstart;nfile<=fend;nfile++)*/

    for(i=0; i<npoints; i++) {
        printf("%f , %fn", Power[i], Temp1[i]);
    }
    getch();

    /***/
    /* Screen graphics: Based on SAMPLE10.C */
    /***/

    /* Initialize chart parameters to defaults */
    gbInitChart(&ch, INIT_ALL, COLOR_SCREEN);

    /* Set up chart input */
    ch.ChartType = XY_CH;
    ch.Npts = npoints;
    ch.Title[0] = Title;

    ch.Title[1] = sn;
    ch.ChartOptions.Xaxis.ttl = TimeTxt;
    ch.Xseries.Data = Time;
    ch.Xseries.DataType = DOUBLEtp;

    ch.Yseries[0].Name = PowerTxt;
    ch.Yseries[0].Data = Power;
    ch.Yseries[0].DataType = DOUBLEtp;

    ch.Yseries[1].Name = Temp1Txt;
    ch.Yseries[1].Data = Temp1;
    ch.Yseries[1].DataType = DOUBLEtp;

```

```

ch.Yseries[2].Name = Temp2Txt;
ch.Yseries[2].Data = Temp2;
ch.Yseries[2].DataType = DOUBLEtp;

/* Enter Graphics Mode */
initgraph(&drv, &mode, bgipath);
if (graphresult() != 0) exit(1);

/* Draw the chart */
ierr = gbChart(&ch);
if (!ierr) bioskey(0);

/* Return to Text Mode */
closegraph();
if (ierr) printf("%s\n", gbErrorMsg(ierr));

}

end:  fputs("stop - no more files found", fpo);
      fclose(fpo);
      puts("Plotting on Screen done!");
return 0;

}

```

Appendix 5

Data Processing Using C Program

Typical Plots

NASA ROMPS Flight Data

Processed for Sample 101

